Implementation of Tuy's cone-beam inversion formula

This content has been downloaded from IOPscience. Please scroll down to see the full text.
(http://iopscience.iop.org/0031-9155/39/3/014)

View the table of contents for this issue, or go to the journal homepage for more

Download details:
IP Address: 155.98.164.38
This content was downloaded on 26/05/2015 at 17:37

Please note that terms and conditions apply.
Implementation of Tuy's cone-beam inversion formula

G L Zeng, R Clack and G T Gullberg
Department of Radiology, Medical Imaging Research Laboratory, University of Utah, Salt Lake City, UT 84132, USA

Received 13 October 1993

Abstract. Tuy's cone-beam inversion formula was modified to develop a cone-beam reconstruction algorithm. The algorithm was implemented for a cone-beam vertex orbit consisting of a circle and two orthogonal lines. This orbit geometry satisfies the cone-beam data sufficiency condition and is easy to implement on commercial single photon emission computed tomography (SPECT) systems. The algorithm, which consists of two derivative steps, one rebinning step, and one three-dimensional backprojection step, was verified by computer simulations and by reconstructing physical phantom data collected on a clinical SPECT system. The proposed algorithm gives equivalent results and is as efficient as other analytical cone-beam reconstruction algorithms.

1. Introduction

In 1983 Tuy presented the first exact cone-beam inversion formula for the reconstruction of a three-dimensional object from x-ray cone-beam data where the orbit of the focal point describes a bounded curve. Over the last 10 years, Tuy’s formulation has never been implemented numerically, whereas, during this time, two other cone-beam algorithms (Smith 1985, Grangeat 1987) have been developed and have since been evaluated through numerical experiments (Grangeat 1991, Smith and Chen 1992, Kudo and Saito 1990, Weng et al 1993). In revisiting Tuy’s inversion formula, we have been able to reformat his formulation so that it leads to an efficient cone-beam reconstruction algorithm.

Tuy’s, Smith’s, and Grangeat’s cone-beam inversion formulas are exact if certain data sufficiency conditions are satisfied. The cone-beam data sufficiency condition requires that every plane that passes through the imaging field of view must also cut through the orbit at least once. In addition, Tuy’s formula also requires that the plane should not be tangent to the orbit at the intersection. For some orbits a plane that passes through a reconstruction point can intersect the orbit of the cone-beam vertex more than once. This may give redundant measurements. In Tuy’s original formula the redundant projections were not used. In our implementation, a new weighting scheme is developed so that all measurements are used by accurately averaging over multiply measured projections.

In this paper, we show the derivation of our cone-beam reconstruction algorithm starting with Tuy’s inversion formula. The algorithm consists of two derivative steps (one derivative with respect to an orbit parameter and one derivative with respect to the projection spatial variable), one rebinning step, and one three-dimensional Radon backprojection step. The algorithm was verified using computer simulated data and cone-beam projection data acquired from a Picker PRISM 2000 two-detector single photon emission computed tomography (SPECT) system using a cone-beam vertex orbit of one circle and two orthogonal lines (figure 1). This orbit (Zeng and Gullberg 1992) satisfies the cone-beam data sufficiency
condition and is easy to implement on a clinical SPECT system. Our implementation of Tuy’s cone-beam formula can be tailored to any differentiable orbit (such as a helical orbit) that satisfies the cone-beam data sufficiency condition.

2. Theory

2.1. Tuy’s cone-beam inversion formula

The cone-beam geometry is shown in figure 2. The focal-point trajectory is referred to as the orbit, which is denoted by $\Phi(\lambda)$ or simply $\Phi$. The focal length $D$ is the distance between the focal point and the axis of rotation. The detector plane is assumed to be at the axis of rotation. The object density function is $f(x)$, where $x$ is a vector in the $x$-$y$-$z$ coordinate system. The modified cone-beam projection of $f(x)$ along the direction of $\Psi/\|\Psi\|$ at the focal point location $\Phi(\lambda)$ is defined as

$$g(\Psi, \Phi) = \int_0^\infty f(\Phi + t\Psi) \, dt = \frac{1}{\|\Psi\|} \int_0^\infty f \left( \Phi + t \frac{\Psi}{\|\Psi\|} \right) \, dt = \frac{1}{\|\Psi\|} g \left( \frac{\Psi}{\|\Psi\|}, \Phi \right)$$

where $\Psi$ is the three-dimensional vector shown in figure 2.

Tuy (1983) derived an inversion formula to reconstruct the object $f(x)$ from the cone-beam projections $g(\Psi/\|\Psi\|, \Phi)$. First consider the three-dimensional Fourier transform of $g(\Psi, \Phi)$ with respect to $\Psi$, which is given as

$$G(\beta, \Phi) = \int \int_{\mathbb{R}^3} g(\Psi, \Phi) e^{-2\pi i \Psi \cdot \beta} \, d\Psi$$

where $\beta$ is a vector in $\mathbb{R}^3$. In this paper, we are only interested in those $G(\beta, \Phi)$ where $\beta$ is a unit vector. Then Tuy’s cone-beam inversion formula is given by

$$f(\alpha) = \int \int_S \frac{1}{2\pi i \Phi'(\lambda)} \frac{\partial}{\partial \lambda} G(\beta, \Phi(\lambda)) \, d\beta$$
where $S$ is the unit sphere, and the orbit parameter $\lambda$ is chosen such that $\beta \cdot x = \beta \cdot \Phi(\lambda)$ and $\beta \cdot \Phi'(\lambda) \neq 0$. The equation $\beta \cdot x = \beta \cdot \Phi(\lambda)$ means that both the point $x$ and the focal point $\Phi(\lambda)$ are in the same plane with the normal vector $\beta$, while $\beta \cdot \Phi'(\lambda) \neq 0$ implies that the orbit is not tangent to the plane. The existence of $\lambda$ is ensured if the data sufficiency condition is satisfied.

2.2. Modification of Tuy's formulation

It has been conjectured that (3) has no practical numerical implementation (Grangeat 1990). In this section, (3) will be modified so that a numerical implementation is possible.

First, let us use (1) to simplify (2) as follows

$$iG(\beta, \Phi) = i \int \int \int_{\mathbb{R}^3} g(\Psi, \Phi) e^{-2\pi i \Psi \cdot \beta} d\Psi$$

$$= i \int \int \int_{0}^{\infty} \frac{1}{r} g(\theta, \Phi) e^{-2\pi i r \theta \cdot \beta} r^2 dr d\theta$$

where $r$ is the spherical radial coordinate, $\theta$ is a three-dimensional unit vector such that $\Psi = r \theta$ and $\|\theta\| = 1$. Grouping terms, we have

$$iG(\beta, \Phi) = \int \int g(\theta, \Phi) \left( \int_{0}^{\infty} i r e^{-2\pi i r \theta \cdot \beta} dr \right) d\theta$$

$$= \frac{1}{2} \int \int g(\theta, \Phi) \left( \int_{-\infty}^{\infty} i r e^{-2\pi i r \theta \cdot \beta} dr \right) d\theta$$

$$+ \frac{i}{2} \int \int g(\theta, \Phi) \left( \int_{-\infty}^{\infty} |r| e^{-2\pi i r \theta \cdot \beta} dr \right) d\theta.$$
The bracket in the first term of (7) is a derivative filter. The bracket in the second term is the common ramp filter used in computed tomography. In (7), the first term is real and odd, and the second term is imaginary and even. We can ignore the second term in (7) because the combination of this term and \( \Phi'(\lambda) \cdot \beta \), which is odd in \( \beta \), will give an odd function when (7) is substituted into (3). This term will vanish when we perform the integration over \( \beta \). Therefore (7) reduces to

\[
2G(\beta, \Phi) = -\frac{1}{4\pi} \int_s g(\theta, \Phi) \delta'(\theta \cdot \beta) \, d\theta + \frac{i}{2} \text{(ignored)}. \tag{8}
\]

Substituting the first term of (8) into (3) and changing the order of the partial derivative and integration, (3) becomes

\[
f(x) = \frac{1}{8\pi^2} \int_s \frac{1}{\Phi'(\lambda) \cdot \beta} \int_s \frac{\partial}{\partial \lambda} g(\theta, \Phi(\lambda)) \delta'(\theta \cdot \beta) \, d\theta \, d\beta. \tag{9}
\]

Accounting for the multiple measurements, (9) can be rewritten as

\[
f(x) = \frac{1}{8\pi^2} \int_s \frac{1}{n(x, \beta)} \sum_{i=1}^{n(x, \beta)} \frac{1}{\Phi'(\lambda_i) \cdot \beta} \int_s \frac{\partial}{\partial \lambda} g(\theta, \Phi(\lambda_i)) \delta'(\theta \cdot \beta) \, d\theta \, d\beta \tag{10}
\]

where \( \lambda_1, \lambda_2, \ldots, \lambda_{n(x, \beta)} \) are solutions to \( \beta \cdot x = \beta \cdot \Phi(\lambda) \) and \( \beta \cdot \Phi'(\lambda) \neq 0 \). Here, \( n(x, \beta) \) is the number of solutions. For the circle-and-line orbit, \( n(x, \beta) \) takes the values of 1, 2, 3, or 4.

2.3. An algorithm implementing Tuy's formula

In this section, we show how to implement (10) for a circle-and-line orbit (figure 1). The \( x-y-z \) coordinate system in figure 2 is set up such that the \( z \)-axis coincides with the axis of rotation for the circular orbit and the \( x-y \) plane contains the circular orbit. We assume that as the detectors move, the face of each detector is always parallel to the tangent of the orbit.

**Step 1.** Evaluate \( (\partial/\partial \lambda) \Phi g(\theta, \Phi(\lambda)) \), for all views \( \Phi \) and all projection samples \( \theta \).

Here, the parameter \( \lambda \) is the arc length of the orbit. For a fixed \( \theta \), we use the following approximation formula (Beyer 1987):

\[
g'(x) \approx g(x - 1.5)/27 - g(x - 0.5) + g(x + 0.5) - g(x + 1.5)/27 \tag{11}
\]

to calculate \( (\partial/\partial \lambda) \Phi g(\theta, \Phi) \) (note that in applying the general form of the derivative in the book by Beyer (1987), the step size \( h \) is chosen to be 0.5). Let \( \Phi_k \) denote \( \Phi(k\lambda) \), where \( \lambda \) is the arc length step size along the orbit.

For the linear portion of the orbit shown in figure 3, \( g(\theta, \Phi_{k-2}) \), \( g(\theta, \Phi_{k-1}) \), \( g(\theta, \Phi_k) \), and \( g(\theta, \Phi_{k+1}) \) are at the same location on the cone-beam detection plane. Using (11), the derivative \( (\partial/\partial \lambda) g(\theta, \Phi) \) is given by

\[
(\partial/\partial \lambda) g(\theta, \Phi) \approx \frac{1}{27} g(\theta, \Phi_{k-2}) - g(\theta, \Phi_{k-1}) + g(\theta, \Phi_k) - \frac{1}{27} g(\theta, \Phi_{k+1}) \tag{12}
\]
Implementation of cone-beam formula

where $\Phi$ is the mid-point between $\Phi_{k-1}$ and $\Phi_k$ on the linear orbit.

For the circular orbit, the locations of $g(\theta, \Phi_{k-2})$, $g(\theta, \Phi_{k-1})$, $g(\theta, \Phi_k)$ and $g(\theta, \Phi_{k+1})$ on the detection plane are different as shown in figure 4, and are denoted as $g_{k-2}(\theta_{k-2})$, $g_{k-1}(\theta_{k-1})$, $g_k(\theta_k)$, and $g_{k+1}(\theta_{k+1})$ in local coordinate systems. The rays defined by $\theta_{k-2}$, $\theta_{k-1}$, $\theta_k$, and $\theta_{k+1}$ are parallel. The unit vectors $\theta_{k-2}$, $\theta_{k-1}$, $\theta_k$, and $\theta_{k+1}$ are defined in their local coordinate systems as shown in figure 4. Similar to (12), we have

$$\frac{\partial}{\partial \lambda}g(\theta, \Phi) = g_{k-2}(\theta_{k-2})/27 - g_{k-1}(\theta_{k-1}) + g_k(\theta_k) - g_{k+1}(\theta_{k+1})/27$$

where $\Phi$ is the mid-point between $\Phi_{k-1}$ and $\Phi_k$ on the circular orbit. Let $\xi$ be the longitudinal coordinate of vector $\theta$ and $\xi_k$ be the longitudinal coordinate of vector $\theta_k$. Likewise, let $\eta$ be the azimuthal coordinate of vector $\theta$ and $\eta_k$ be the azimuthal coordinates of vector $\theta_k$. We have

$$\eta = \eta_{k-2} = \eta_{k-1} = \eta_k = \eta_{k+1}$$

(14)

$$\xi_k = \xi + 0.5\gamma$$

(15)

$$\xi_{k-1} = \xi - 0.5\gamma$$

(16)

$$\xi_{k+1} = \xi + 1.5\gamma$$

(17)

$$\xi_{k-2} = \xi - 1.5\gamma$$

(18)

$$\gamma = 2\pi/\text{number of angles}.$$  

(19)

Linear interpolations are used to approximate the values of $g_{k-2}(\theta_{k-2})$, $g_{k-1}(\theta_{k-1})$, $g_k(\theta_k)$, and $g_{k+1}(\theta_{k+1})$.

Step 2. Evaluate $\int f_2(\partial/\partial \lambda)g(\theta, \Phi)\delta'(\theta - \beta) \, d\theta$, for all $\Phi$ and all $\beta$.

At each fixed $\Phi$, we divide the set of $\beta$ into subsets $S_\alpha$, which consist of unit vectors $\beta$ in the plane that contains the focal point $\Phi$ and the $t$-axis. The $t$-axis is defined on the detection plane and has an angle $\alpha$ with the $z$-axis (the axis of rotation). For the set $S_\alpha$, we define a

![Figure 3. Projection points $\theta$ used for evaluating the derivative of the linear-orbit projection data with respect to the arc length $\lambda$. All rays in each local coordinate system are parallel and are in the direction of the global vector $\theta$. Therefore projection points are at the same location in each view.](image-url)
Figure 4. Projection points \( \theta \) used for evaluating the derivative of the circular-orbit projection data with respect to the arc length \( \lambda \). All rays in each local coordinate system are parallel and are in the direction of the global vector \( \theta \). The rays in each local coordinate system, \( \theta_{k-2}, \theta_{k-1}, \theta_k, \) and \( \theta_{k+1} \), are parallel, but the \( \xi \) values are different. (In this figure, we move the detector closer to the focal point intentionally.)

\( (u, t)_{x, \alpha} \) coordinate system as shown in figure 5. We evaluate \( \int \frac{\partial}{\partial \lambda} g(\theta, \Phi) \delta'(\theta \cdot \beta) \, d\theta \) for each \( S_{\alpha} \) separately.

For each \( S_{\alpha} \), we set up a local coordinate system \( t-u-v \) as shown in figure 5. The purpose of this step is to change the variables into the \( t-u-v \) system. The origin of the local coordinate system is the same as the origin in the \( x-y-z \) system. The \( u \)-axis points toward the focal point \( \Phi(\lambda) \). Both the \( t \)-axis and the \( u \)-axis are on the detector plane. The angle between the \( t \)-axis and the \( z \)-axis is \( \alpha \). Let \( \gamma_\beta \) be the angle between the unit vector \( \beta \) and the \( t' \)-axis. The \( t' \)-axis is parallel to the \( t \)-axis and the focal point \( \Phi(\lambda) \) is on the \( t' \)-axis. In the \( t-u-v \) system, the unit vector \( \beta \in S_{\alpha} \) and the unit vector \( \theta \) can be expressed as

\[
\beta = (\cos \gamma_\beta, 0, \sin \gamma_\beta)
\]

\[
\theta = \left(\frac{t}{\sqrt{t^2 + u^2 + D^2}}, \frac{u}{\sqrt{t^2 + u^2 + D^2}}, -\frac{D}{\sqrt{t^2 + u^2 + D^2}}\right)
\]

respectively. We also have

\[
d\theta = \left[\frac{\sqrt{t^2 + u^2} / (t^2 + u^2 + D^2)^{3/2}}{}\right] \, dt \, du.
\]

Therefore

\[
\int \int \frac{\partial}{\partial \lambda} g(\theta, \Phi) \delta'(\theta \cdot \beta) \, d\theta
\]

\[
= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \frac{\partial}{\partial \lambda} \left[\theta(t, u), \Phi\right] \left[\frac{\sqrt{t^2 + u^2}}{(t^2 + u^2 + D^2)^{3/2}} \delta'(\frac{t \cos \gamma_\beta - D \sin \gamma_\beta}{\sqrt{t^2 + u^2 + D^2}})\right] \, dt \, du.
\]

Using the derivative property of the Dirac delta function (Sneddon 1972)

\[
\int_{-\infty}^{\infty} f(x) \delta'(ax) \, dx = \int_{-\infty}^{\infty} f(x) \frac{1}{a^2} \delta'(x) \, dx = -\frac{f'(0)}{a^2}
\]
Figure 5. A coordinate system \((u, t)_{\kappa,\ell}\) is set up such that the \(t\)-axis is the intersection of the detector plane and \(S_\kappa\). For \(\beta \in S_\kappa\), the projection points defined by all \(\theta\) with \(\beta \cdot \theta = 0\) form a line perpendicular to the \(t\)-axis. Here the axes \(t, u, w,\) and \(z\) are in the same detector plane. The plane \(P\) contains the focal point \(\Phi\) and the \(t\)-axis. The \(t'\)-axis is parallel to the \(t\)-axis and is in the plane \(P\).
we have
\[ \int \int_S \frac{\partial}{\partial \lambda} g(\theta, \Phi) \delta'(\theta \cdot \beta) \, d\theta \]
\[ = - \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \frac{1}{\cos^2 \gamma_\beta} \frac{\partial}{\partial \lambda} g(\theta(t, u), \Phi) \frac{\sqrt{t^2 + u^2}}{\sqrt{t^2 + u^2 + D^2}} \delta'(t - D \tan \gamma_\beta) \, dt \, du \]
\[ = - \int_{-\infty}^{\infty} \frac{1}{\cos^2 \gamma_\beta} \frac{d}{dt} \left[ \frac{\partial}{\partial \lambda} g(\theta(t, u), \Phi) \frac{\sqrt{t^2 + u^2}}{\sqrt{t^2 + u^2 + D^2}} \right] \bigg|_{t = D \tan \gamma_\beta} \, du. \quad (24) \]

Let \( t_\beta = D \tan \gamma_\beta \), then \( 1/\cos^2 \gamma_\beta = (D^2 + t_\beta^2)/D^2 \). Equation (24) becomes
\[ \int \int_S \frac{\partial}{\partial \lambda} g(\theta, \Phi) \delta'(\theta \cdot \beta) \, d\theta = \int_{-\infty}^{\infty} \hat{g}(u, t_\beta) \frac{D^2 + t_\beta^2}{D^2} \, du \triangleq A_{\Phi, \alpha}(t_\beta) \quad (25) \]
where
\[ \hat{g}(u, t_\beta) = - \frac{d}{dt} \left[ \frac{\sqrt{t^2 + u^2}}{\sqrt{t^2 + u^2 + D^2}} \frac{\partial}{\partial \lambda} g(\theta(t, u), \Phi) \right] \bigg|_{t = t_\beta}. \quad (26) \]

Equation (26) is a directional derivative and can be evaluated by
\[ \hat{g}(u, t) = - \sin \alpha \frac{\partial}{\partial w} \left\{ \frac{\sqrt{t^2 + u^2}}{\sqrt{t^2 + u^2 + D^2}} \frac{\partial}{\partial \lambda} [g(\theta, \Phi)] \right\} - \cos \alpha \frac{\partial}{\partial z} \left\{ \frac{\sqrt{t^2 + u^2}}{\sqrt{t^2 + u^2 + D^2}} \frac{\partial}{\partial \lambda} [g(\theta, \Phi)] \right\} \quad (27) \]
where each partial derivative is calculated via (11). Equation (27) can be executed once for all \( \beta \in S_\alpha \). Equation (25) is a parallel line integral with a weighting function, and can be executed once for all \( \beta \in S_\alpha \) (i.e. for all \( t \)).

In step 2, the direction \( \beta \) (or \( t \)) is tested for whether the projection data are truncated for the linear orbit. If truncation occurs, the truncated \( A_{\Phi, \alpha}(t) \) is discarded as illustrated in figure 6.

**Step 3.** Evaluate \((1/\Phi' \cdot \beta) \int_S \frac{\delta}{\delta X} g(\theta, \Phi) \delta'(\theta \cdot \beta) \, d\theta\), which is \((1/\Phi' \cdot \beta) A_{\Phi, \alpha}(t)\).

The variable \( t \) depends on \( \beta \), therefore \( \Phi' \cdot \beta \) is different for different \( t \). We use \( \gamma_\beta = \tan^{-1}(t/D) \) to determine the unit vector \( \beta \), which is used to evaluate the inner product \( \Phi' \cdot \beta \). If \( \Phi' \cdot \beta = 0 \), \((1/\Phi' \cdot \beta) A_{\Phi, \alpha}(t) \) is discarded, because it is not included in (10).

**Step 4.** Rebin 1D arrays into the 3D spherical system.

For each \( \Phi \), the 1D array \((1/\Phi' \cdot \beta) A_{\Phi, \alpha}(t)\) is embedded (rebinned) into a 3D data array \( R(\rho, \beta) \) where \( \beta \) has two indices, and \( \rho = \Phi' \cdot \beta = \beta \cdot \alpha \). Here, \( \rho \) can be evaluated by \( \rho = t/\cos \gamma_\beta = t D/\sqrt{D^2 + t^2} \) (see figure 5). At the same time, a value of one is added to a 3D counting array \( C(\rho, \beta) \) with the same indices. After this step is finished for all \( \Phi, n(\alpha, \beta) = C(\rho, \beta) \). Linear interpolations are needed in this step.

**Step 5.** Evaluate \([1/n(\alpha, \beta)] \sum_{i=1}^{n(\alpha, \beta)} (1/\Phi' \cdot \beta) \int_S g(\theta, \Phi(\lambda_i)) \delta'(\theta \cdot \beta) \, d\theta\).

The 3D data array \( R(\rho, \beta) \) is divided by the 3D counting array \( C(\rho, \beta) \) point by point, obtaining \( D(\rho, \beta) \).

**Step 6.** Evaluate \( f(\alpha) = (1/8\pi^2) \int \int_S D(\alpha \cdot \beta, \beta) \, d\beta \).

This operation is accomplished by performing the 3D Radon backprojection.
3. Methods

The circle-and-line orbit must first be specified in order that the data sufficiency condition is satisfied for the object that is being reconstructed. We assume that the object is contained within a sphere of radius $R$. An expression for the half length $L$ of the linear orbit in figure 7 can be derived from simple geometric considerations:

$$L/R = D / \sqrt{D^2 - R^2}.$$  \hspace{1cm} (28)

In order to satisfy the cone-beam data sufficiency condition for the spherical object illustrated in figure 7, the circular orbit must have a radius $R$, and $L$ must satisfy

$$L \geq DR / \sqrt{D^2 - R^2}.$$  \hspace{1cm} (29)

Figure 6. When the detector translates, the object may be truncated. The truncated line integrals are discarded, and the non-truncated line integrals are used for image reconstruction.

Figure 7. The length $2L$ of the linear orbits is determined by the phantom radius $R$ and the focal length $D$. 
3.1. Computer simulation

A 3D mathematical phantom, shown in figure 8, was used to verify our algorithm. The phantom contained five flat ellipsoids of equal density. All ellipses had axes of 20, 20, and 2.5 voxels in the x-, y-, and z-directions. The distances between the centres of the two adjacent ellipsoids were eight voxels. The central ellipsoid was positioned at the centre of the circular orbit plane, which was in the x–y plane. Figure 8(b) shows a noise-free sagittal cut of the phantom generated by the computer.

![Computer simulation](image)

Figure 8. Computer simulation: (a) phantom dimensions and the phantom orientation relative to the axis of rotation z; (b) the ideal sagittal cut of the phantom.

Projections were generated from analytical line integrals of the phantom shown in figure 8. Attenuation, collimator geometric response, and scatter were not simulated. The cone-beam focal length \( D \), and the radius of the circular orbit, were both equal to 180 voxels. For computer simulations, it was assumed that the detectors were at the centre of rotation of the circular orbit and centrally located between the two physical detectors in figure 1 for the linear orbit. The phantom was contained within a sphere of radius \( R = 25.6 \) voxels. According to (29), the half length \( L \) of the linear orbit should be greater than or equal to 25.87 voxels. We choose \( L = 26 \) voxels for our simulation. For the circular orbit,
Implementation of cone-beam formula

120 64 × 64 projections were generated over 360°. For the linear orbit, projections were generated for two opposing detectors. Each detector had one projection at the central orbit and five projections on either side for a total of 11 linear projections for each detector.

The simulated projections were reconstructed into a 64 × 64 × 64 array. The voxel size was equal to the projection pixel size. Sagittal cuts in the z-plane through the reconstructed phantom were displayed and analysed for slice-to-slice cross talk.

3.2. Phantom study

A physical Defrise phantom (Data Spectrum Inc., Durham, NC) was also used to evaluate our algorithm (figure 9). The phantom contains six flat hot discs. All discs have a radius of 11 cm, a thickness of 1.1 cm, and separation of 2.3 cm. The phantom was positioned on a bed so that the long axis of the phantom was aligned along the z-axis as shown in figure 9.

A Picker 2000 two-detector SPECT system was used to acquire the cone-beam projections. The focal length of each cone-beam collimator was 65 cm. The radius of the detector orbit was 17 cm. For the circular orbit, 120 views were sampled over 360°. The linear orbit was 20 cm long, 10 cm on each side of the central plane of the circular orbit. A total of 11 projections was acquired along the linear orbit, similar to the computer simulations. The projections were acquired into 64 × 64 arrays with a pixel size of 0.934 cm at the detector. In some of the views along the linear orbit, the projection data were truncated.

The projections were reconstructed into a 64 × 64 × 64 array with a voxel dimension of 0.67 cm. The reconstruction method was exactly the same as that used for the computer simulations. The truncation problem was handled as described in step 3 of section 2.3.

3.3. Algorithm implementation

We discretized the cone-beam projections in 142 64 × 64 arrays. The circular orbit had 120 views uniformly distributed over 360°. Each linear orbit had 11 views with uniform spacing between two adjacent views. For each view, we did the following:

(i) calculated the derivative of the projection data according to (12) and (13) for the linear and circular orbits, respectively, obtaining a set of 2D data in 142 projection planes. The derivative was approximated by the four-point finite difference;
(ii) computed the directional derivatives for each of the 2D projection planes at 130 angles uniformly distributed from 0 to \( \pi \), using (27);

(iii) computed parallel projections on each of the 2D projection planes at the same 130 angles as in (ii), obtaining 130 1D arrays for each view. In each 1D array, there were 64 values of \( t \). Before each parallel projection, the data are weighted by a factor \( (D^2 + t^2)/D^2 \).

(iv) weighted each point in the 1D arrays by \( 1/(\Phi' \cdot \beta) \). If \( \Phi' \cdot \beta = 0 \), this data point was discarded;

(v) let \( \rho = \Phi \cdot \beta \) and let the longitude and azimuthal coordinates of \( \beta \) be \( \phi_\beta \) and \( \theta_\beta \), respectively. The data points from step (iv) were rebinned into \( R(\rho, \phi_\beta, \theta_\beta) \) in Radon space. Here, \( \rho \) was discretized into integers from \(-32\) to \(32\), \( \phi_\beta \) was discretized into 64 values from 0 to \( 2\pi \), and \( \theta_\beta \) was discretized into 40 values from 0 to \( \pi/2 \). At the same time, a value of one was added to the counting array \( C(\rho, \phi_\beta, \theta_\beta) \) with the same indices. Linear interpolations were needed in this step;

(vi) performed backprojection of the array \( R(\rho, \phi_\beta, \theta_\beta)/C(\rho, \phi_\beta, \theta_\beta) \) into a \( 64 \times 64 \times 64 \) image volume. The two-stage 2D backprojection approach was used to backproject the 3D Radon data (Marr et al 1981).

4. Results

4.1. Computer simulation

The cone-beam reconstruction algorithm was coded in C on a SUN SPARC 2 workstation. The total computer time for a \( 64 \times 64 \times 64 \) reconstruction was approximately 5 min. Figure 10(a) shows reconstruction from only the circular-orbit data using our algorithm based on Tuy’s formula, and figure 10(b) shows reconstruction from the complete circle-and-line-orbit data. It is observed that the reconstruction without the linear-orbit data is much worse than the reconstruction with the circle-and-line-orbit data. We see from figure 10(a) that the reconstruction from only the circular-orbit data has severe cross-talk artifacts between the discs in the \( z \)-direction. The top and bottom discs have lower intensity than the central disc, and the cross-talk artifacts are distributed between the gaps of the discs. Using data from both the circular and linear orbits satisfies the cone-beam data sufficiency condition and the reconstruction is much improved as shown in figure 10(b).

4.2. Phantom study

Figure 11 shows the reconstructions for the phantom study. Even though the projections were attenuated no attenuation correction was incorporated. Again a better reconstruction is obtained reconstructing the circle-and-line-orbit data than only reconstructing the circular-orbit data.

5. Conclusions

A cone-beam reconstruction algorithm for a circle-and-line orbit was derived from Tuy’s cone-beam inversion formula. The major modifications to Tuy’s formula are (i) not using the Fourier transform to calculate the function \( G \), but instead using parallel projections followed by 1D derivatives, (ii) not using the real part of the function \( G \) because it has no effect upon the final image, and (iii) appropriately normalizing for multiply measured projections. We find from our computer simulations that our implementation of Tuy’s inversion formula is
Implementation of cone-beam formula

Figure 10. Computer simulation: (a) reconstruction with only circular-orbit data using the algorithm based on Tuy’s formula; (b) reconstruction with complete circle-and-line-orbit data using the algorithm based on Tuy’s formula.

as accurate and as efficient as our implementation of Smith’s (Zeng and Gullberg 1992) and Grangeat’s (Weng et al 1993) cone-beam reconstruction algorithms.

In implementing the algorithm, the whole object should be within the field of view as the camera rotates through the circular orbit. When the camera moves through the linear orbit, portions of the object may project outside the field of view. These truncated data are redundant and not needed for the reconstruction. This situation is properly handled by the algorithm. However, we realize that too much truncation is detrimental since it represents time wasted in the scanning of the object when we consider the overall sensitivity of the scanning protocol.

This circle-and-line orbit is easy to implement on present clinical two-detector SPECT systems. One needs only to rotate each detector by 180° followed by a linear translation of the camera or patient bed. The reconstruction algorithm can be easily tailored to other orbit geometries such as the helical orbit. The only part of the algorithm that needs to be modified is the evaluation in step 1 of derivatives of the projection data over the orbit.
Figure 11. Physical phantom study: (a) reconstruction with only circular-orbit data using the algorithm based on Tuy's formula; (b) reconstruction with complete circle-and-line-orbit data using the algorithm based on Tuy's formula.

From the viewpoint of implementation, Tuy's formula is sensitive to the orbit shape and the sampling interval on the orbit. If the orbit is not differentiable, Tuy's formula does not apply, while Grangeat's and Smith's algorithms are still appropriate. Another drawback of Tuy's formula is that the approximation of the derivative of the projections over the orbit may have a large error term, if the orbit sampling interval is relatively large.

Future research needs to consider developing a cone-beam reconstruction algorithm without the rebinning step and optimizing the sampling strategy.

Acknowledgments

The research work presented in this manuscript was partially supported by the Whitaker Foundation, NIH Grant RO1 HL 39792, and Picker International. We thank Paul Christian for the phantom data acquisition. We would also like to thank the Biodynamics Research Unit, Mayo Foundation for use of the ANALYZE software package.
References


Grangeat P 1987 Analysis d'un système d'imagerie 3D par reconstruction à partir de X en géométrie conique PhD Thesis l'Ecole Nationale Supérieure Des Télécommunications


Marr R B, Chen C N and Lauterbur 1981 On two approaches to 3D reconstruction in NMR tomography *Mathematical Aspects of Computed Tomography* (Lecture Notes in Medical Informatics 8) ed G T Herman and F Natterer (Berlin: Springer) pp 225–40


