

Distance-driven projection and backprojection in three dimensions

Bruno De Man and Samit Basu

CT Systems and Applications Laboratory, GE Global Research, Niskayuna, NY, USA

E-mail: bruno.deman@research.ge.com and samit.basu@research.ge.com

Received 27 October 2003

Published 19 May 2004

Online at stacks.iop.org/PMB/49/2463

DOI: 10.1088/0031-9155/49/11/024

Abstract

Projection and backprojection are operations that arise frequently in tomographic imaging. Recently, we proposed a new method for projection and backprojection, which we call *distance-driven*, and that offers low arithmetic cost and a highly sequential memory access pattern. Furthermore, distance-driven projection and backprojection avoid several artefact-inducing approximations characteristic of some other methods. We have previously demonstrated the application of this method to parallel and fan beam geometries. In this paper, we extend the distance-driven framework to three dimensions and demonstrate its application to cone beam reconstruction. We also present experimental results to demonstrate the computational performance, the artefact characteristics and the noise-resolution characteristics of the distance-driven method in three dimensions.

1. Introduction

Computed tomography (CT) reconstruction is the process of recovering n -dimensional (nD) image data from a set of integrals of that data over lower-dimensional subspaces. We consider the case in which the integrals are over lines (one-dimensional), a case sometimes referred to as the x-ray transform. There are numerous, well-established examples of the x-ray transform in medical imaging (Suetens 2002), including computed tomography (CT), single photon emission computed tomography (SPECT), positron emission tomography (PET), tomosynthesis and projection mode 2D magnetic resonance imaging (MRI). In many of these applications a forward and/or a reverse model of the acquisition process is required for image reconstruction, artefact correction, or simulation purposes. The forward model, in which line integrals of a known image are calculated, is known as the *projection* operation. The reverse model, generally defined as the transpose (or adjoint) of the forward model, is known as the *backprojection* operation.

The most prevalent application of the backprojection operation is in the filtered backprojection (FBP) reconstruction algorithms, which are based on inversion formulae for the x-ray transform (Suetens 2002, pp 77–80). The filtering stage consists of the application of a *ramp* filter, either in the spatial or in the frequency domain, often combined with some type of low-pass filter to suppress noise and aliasing artefacts. The second stage is simply a (weighted) backprojection.

The projection operation arises in simulation of the physical system or acquisition process. Note that in general, physical models require far more complexity than is expressed by a collection of line integrals. However, many real-world physical effects can be well approximated by taking collections of line integrals and combining them in appropriate ways (De Man *et al* 1999). For example, in CT, finite focal spot sizes and finite detector cell sizes are commonly modelled by taking clusters of point sources and point detectors, the responses of which are combined to simulate the physical beam aperture. For each point source and point detector, the problem of simulation is reduced to line integral calculation. A similar approach works for azimuthal blur (motion blur that occurs when acquisition and rotation take place simultaneously): a number of discrete angular positions, each of which results in a task of computing line integrals, are combined to simulate the blur. When constructing a simulation model, polychromatic effects can be handled through material decomposition approaches or through energy bin discretization (De Man *et al* 2001). If the image is decomposed into a set of images, each of a single material, then line integral projections of these single material images can be combined with the appropriate spectral weights to compute the polychromatic response. Thus, in many practical cases of interest, complex physical forward models can be reduced to the problem of (rapid) line integral calculation.

Another application of projection and backprojection is in iterative reconstruction, in which repeated applications of the forward and reverse model are used to solve for the image that best fits the measurements according to an appropriate objective function, and in iterative correction schemes, an example of which is the reduction of beam hardening artefacts (Hsieh *et al* 2000).

Many methods exist for projection and backprojection, all of which provide some compromise between computational complexity and accuracy. Implementations of backprojection for FBP reconstruction usually use a *pixel-driven* approach (Herman 1980, Peters 1981, Zhuang *et al* 1994). Pixel-driven approaches to backprojection generally require the notion of a regularly spaced detector array. The pixel-driven approach is well suited for hardware implementation, where the high computational complexity often required by the intersection calculation (divisions in the case of a flat detector; arc-tangent calculations in the case of a curved detector) can be handled through specialized circuitry (Kotian *et al* 1995). On general purpose microprocessors, however, pixel-driven approaches tend to suffer from poor performance because of that complexity. Pixel-driven *projection* is a similar process, but now the detector samples are updated with values from the pixels, using similar weights. Simple pixel-driven projection is rarely used because it introduces high-frequency artefacts (Zeng and Gullberg 1993, De Man and Basu 2002). These artefacts can be prevented by using more sophisticated weighting schemes, but only at the expense of increased computational complexity. For example, a technique called *splatting* has been developed for volume rendering applications, which consists of using the analytical footprint of a voxel on the detector (Zwicker *et al* 2002). Its applicability to iterative reconstruction has been demonstrated in Mueller *et al* (1999). To circumvent the high computational complexity of the footprint calculation, some authors pre-calculate and store the projection and backprojection weights (Fessler 1994, p 291). This approach becomes impractical for large datasets, such as in CT.

A fundamentally different approach is taken with the *ray-driven* methods. These methods work by tracing rays through the image, essentially discretizing the line integral directly, and thus approximating each line integral as a weighted sum of all image pixels that lie close to the ideal line. The exact approach used to select and weight the pixels varies with different methods. One approach is to weight the contribution of each pixel by the intersection length of the line with that pixel (Herman 1980, Siddon 1985, Zeng and Gullberg 1993, Zhuang *et al* 1994). Another approach performs linear interpolation between two pixels for each row or column intersected by the projection line (Joseph 1983, Zhuang *et al* 1994). A backprojection operator can be defined as the transpose of this operation, so that pixels close to the line are updated with weighted contributions from that line. Ray-driven methods are generally well-suited for projection, but tend to introduce artefacts (Moiré patterns) in the backprojection (De Man and Basu 2002, 2004). Furthermore, ray-driven methods generally have highly non-sequential memory access patterns, making memory bandwidth a limiting factor in their performance.

Other variants of pixel- and ray-driven approaches have also been suggested, such as methods using nearest-neighbour or no interpolation (Peters 1981, Cho *et al* 1990, Yu and Huang 1993) spherical basis functions (Lewitt 1992, Zhuang *et al* 1994, Mueller *et al* 1999) and natural pixels (Buonocore *et al* 1981, Hsieh *et al* 1996). For projection, other methods from the visualization field can be adopted. The shear-warp volume rendering algorithm is somewhat similar to the distance-driven approaches we present, albeit with more interpolation steps (Lacroute and Levoy 1994). Furthermore, the interpolation steps in shear-warp volume rendering (analogous to projection) are similar to those in ray-driven approaches. Hence, we expect similar difficulties in using it for backprojection.

A totally different class of methods for projection and backprojection is the so-called ‘fast’ algorithms, which exploit properties of the Radon transform or Fan Beam transform to significantly reduce the computational complexity (Basu and Bresler 2000). These algorithms typically require $O(n^2 \log_2 n)$ operations to compute the projection or backprojection of n views with an $n \times n$ image grid, a speed up of roughly $n/\log_2 n$ over conventional approaches. While very preliminary work has shown application of fast algorithms to iterative reconstruction (Basu 2000), these algorithms are generally not flexible enough to capture the wide range of geometries encountered in practice. Their reliance on properties of the Radon or Fan Beam transform means that deviations from these transforms generally invalidate the operation of the algorithms.

In De Man and Basu (2002, 2003), we introduced a novel projection and backprojection approach, which we call *distance-driven*, that has a low arithmetic complexity, a highly sequential (and thus predictable) memory access pattern, and that avoids the artefact characteristics of ray-driven backprojection and pixel-driven projection. 2D distance-driven projection or backprojection works by mapping pixel boundaries and detector boundaries to a common axis, and then applying a one-dimensional kernel operation to map data from one set of boundaries to another. The choice of kernel dictates the properties of the resulting algorithm. A journal paper detailing with the 2D distance-driven method has been submitted for publication.

In this paper, we extend the distance-driven framework to 3D and to cone beam imaging geometries. In section 2 we present the theoretical foundations of distance-driven projection and backprojection, analysing it (along with pixel-driven and ray-driven approaches) in one, two and three dimensions. In section 3 we present numerical experiments that explore performance, artefact level, and noise/spatial resolution characteristics. We also present reconstructions of real measurements demonstrating a reduction in artefact level relative to ray-driven projection/backprojection. Finally, in section 4 we present conclusions.

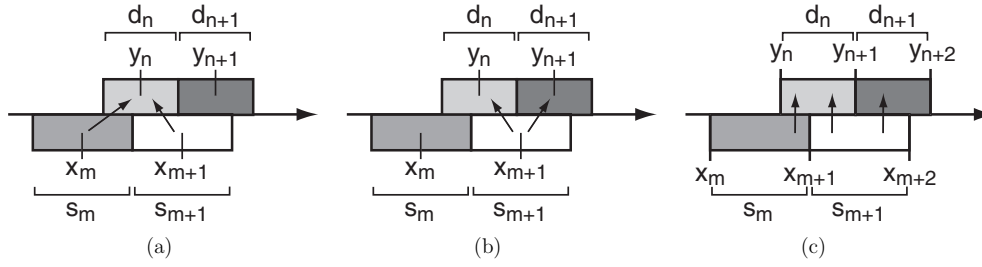


Figure 1. (a) The linear interpolation kernel calculates each destination sample by performing linear interpolation between the two nearest source samples. (b) The transpose of the linear interpolation kernel results in high-frequency artefacts in the destination domain. (c) The distance-driven kernel uses the length of overlap between each source and each destination to perform a weighted sum of the source values.

2. Theory

2.1. One dimension

One way to analyse different projection and backprojection methods is in terms of re-sampling of one-dimensional signals. Suppose that the source signal is defined by a set of sample values s_0, s_1, \dots, s_{M-1} and sample locations $x_0 < x_1 < \dots < x_{M-1}$ and that the destination signal is defined by a set of sample values d_0, d_1, \dots, d_{N-1} and sample locations $y_0 < y_1 < \dots < y_{N-1}$. The re-sampling problem is then a simple matter of determining the destination signal values $\{d_j\}$ from $\{s_i\}$, $\{x_i\}$ and $\{y_j\}$, a process we denote as the *kernel operation*.

Let m and n be the instances of i and j corresponding to the example in figure 1. Figure 1(a) is a schematic representation of the linear interpolation kernel operation, resulting in

$$d_n = \frac{x_{m+1} - y_n}{x_{m+1} - x_m} s_m + \frac{y_n - x_m}{x_{m+1} - x_m} s_{m+1}. \quad (1)$$

Denoting this operation by the system matrix $A = \{a_{i,j}\}$, we have for example:

$$a_{m+1,n} = \frac{y_n - x_m}{x_{m+1} - x_m}.$$

Application of the linear interpolation kernel in the reverse direction, i.e., from $\{d_j\}$ to $\{s_i\}$, would result in

$$s_{m+1} = \frac{y_{n+1} - x_{m+1}}{y_{n+1} - y_n} d_n + \frac{x_{m+1} - y_n}{y_{n+1} - y_n} d_{n+1}. \quad (2)$$

Denoting this reverse operation by the system matrix $B = \{b_{i,j}\}$, we have, for example,

$$b_{m+1,n} = \frac{y_{n+1} - x_{m+1}}{y_{n+1} - y_n}.$$

Because generally $a_{m+1,n} \neq b_{m+1,n}$, operators A and B are not transpose operations.

Figure 1(b) is a schematic representation of the transpose of the linear interpolation kernel (sometimes referred to as *linear anteroprolation*), which results in

$$d_n = \frac{x_m - y_{n-1}}{y_n - y_{n-1}} s_m + \frac{y_{n+1} - x_{m+1}}{y_{n+1} - y_n} s_{m+1}. \quad (3)$$

To define the distance-driven kernel operation, we redefine the sample locations x_i and y_j . We let these sample locations now denote the *boundaries* of the source and destination

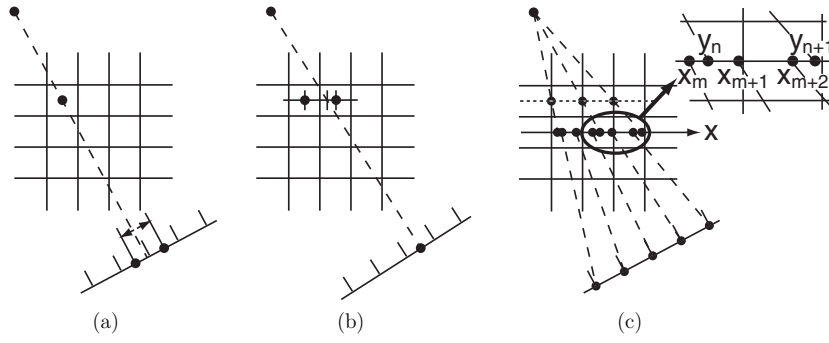


Figure 2. (a) Pixel-driven projection–backprojection works by connecting a line from the focal spot through the centre of the pixel of interest to the detector. (b) Ray-driven projection–backprojection works by connecting a line from the focal spot through the image to the centre of the detector cell of interest. (c) Distance-driven projection–backprojection works by mapping the boundaries of each pixel and detector cell onto a common axis; the length of the overlap is used as (back-)projection weight.

channels, as illustrated in figure 1(c). We define a piecewise constant source signal $s(x)$ with value s_i on the interval $[x_i, x_{i+1})$:

$$s(x) = \sum_i s_i \mathbf{1}_{x_i \leq x < x_{i+1}}.$$

We then define the destination signal d_j in terms of integrals of $s(x)$ over the interval $[y_j, y_{j+1}]$:

$$\begin{aligned} d_j &= \int_{y_j}^{y_{j+1}} s(x) dx = \sum_i s_i \int_{y_j}^{y_{j+1}} \mathbf{1}_{x_i \leq x < x_{i+1}} dx \\ &= \sum_i s_i |[y_j, y_{j+1}] \cap [x_i, x_{i+1})| \end{aligned} \quad (4)$$

where $|\cdot|$ denotes the length of the interval. For the setup in figure 1(c), this results in

$$d_n = (x_{m+1} - y_n) s_m + (y_{n+1} - x_{m+1}) s_{m+1}. \quad (5)$$

The boundary-based kernel has the property that it is self-adjoint because

$$|[y_j, y_{j+1}] \cap [x_i, x_{i+1})| = |[x_i, x_{i+1}) \cap [y_j, y_{j+1}]|.$$

The kernel weights can also be normalized by the width of the source or destination intervals, for example,

$$d_n = \frac{x_{m+1} - y_n}{y_{n+1} - y_n} s_m + \frac{y_{n+1} - x_{m+1}}{y_{n+1} - y_n} s_{m+1}. \quad (6)$$

2.2. Two dimensions

Having defined the different kernel operations in the previous section, it is now straightforward to explain the 2D projection and backprojection methods of interest.

Pixel-driven backprojection (figure 2(a)) works by connecting a line from the focal spot through the centre of the pixel of interest to the detector. Once a location of intersection on the detector is calculated, a value is obtained from the detector by (typically linear) interpolation, and the result is accumulated in the pixel (Herman 1980). This corresponds to the linear

interpolation operation from equation (1), where a sinogram row is the source signal and an image row is the destination signal. For each image row, the pixel centres are mapped onto the detector, followed by the 1D re-sampling operation defined in equation (1). Pixel-driven projection is defined as the transpose operation, corresponding to the transpose kernel from equation (3). In order to correctly model line integrals, pixel-driven projection must incorporate additional weight factors, including the secant of the angle of the projection line as well as the ratio between the pixel size and the detector size. Pixel-driven projection is rarely used since it results in high-frequency artefacts (Zeng and Gullberg 1993, De Man and Basu 2002, 2004).

Ray-driven projection (figure 2(b)) works by connecting a line from the focal spot through the image to the centre of the detector cell of interest. For every image row (or column), a location of intersection is calculated, a value is obtained from the image row, typically by linear interpolation (Joseph 1983), and the result is accumulated in the detector cell. This corresponds to the linear interpolation operation from equation (1), where an image row is the source signal and a sinogram row is the destination signal. For each image row, the detector centres are mapped onto the image row, followed by the 1D re-sampling operation defined in equation (1). Ray-driven backprojection is defined as the transpose operation, corresponding to the transpose kernel from equation (3). Ray-driven backprojection results in high-frequency artefacts (De Man and Basu 2002, 2004).

In De Man and Basu (2002, 2003), we presented a novel projection and backprojection technique, which we call distance-driven, combining the advantages of the ray-driven and pixel-driven methods. To clearly understand the new method, one should first realize that every view (or source position) defines a bijection between the position on the detector and the position within an image row (or column). In other words, every point within an image row is mapped uniquely onto a point on the detector, and vice versa. This allows one to define a length of overlap between each image pixel and each detector cell. To calculate this length of overlap we could map all pixel *boundaries* in an image row of interest onto the detector, or all detector cell *boundaries* onto the centerline of the image row of interest. In practice, we map both sets of boundaries onto a common line, e.g. the x -axis (figure 2(c)). This is achieved by connecting all pixel boundaries and all detector cell boundaries with the source and calculating the x -intercepts. Based on these boundaries, we calculate the length of overlap between each image pixel and each detector cell (as seen on the x -axis) and then use the normalized length of overlap to define the weight used in projection and backprojection (symmetric). This corresponds to applying the distance-driven kernel operation from equation (4) to the mapped boundary locations (zoomed portion of figure 2(c)). This is achieved by performing a loop over all boundaries (both pixel and detector cell boundaries), for instance starting at the boundary with the smallest x intercept and stopping at the boundary with the highest x intercept. The normalization consists of dividing by the pixel width (for FBP) or by the detector width (for simulation). In order to correctly model a line integral, we also need to divide by the cosine of the angle of the projection line, similar to Joseph's method.

Essentially, this technique differs from pixel-driven and ray-driven techniques in two aspects. One aspect is the fact that the main loop is not over image pixels or detector cells, but over x intercepts, hence the name distance-driven. This aspect makes the method fast. The second aspect is the fact that the length of overlap is used, rather than a regular interpolation kernel. This aspect makes the method artefact-free and symmetric, and is responsible for the improved image quality. Hybrid methods exist. For instance, one can map the pixel and detector *centres* on the x -axis and perform linear interpolation. This example would result in a speed improvement, but it would not affect the image quality.

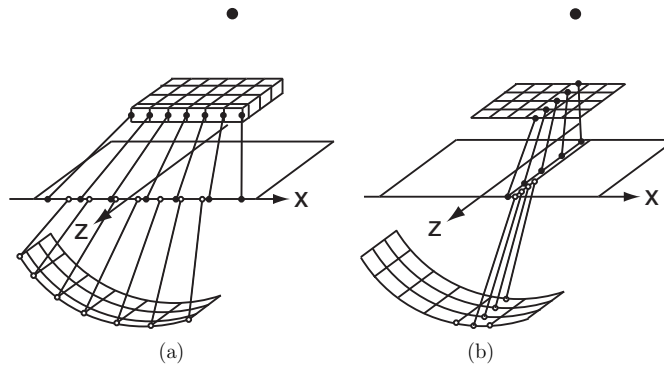


Figure 3. 3D distance-driven method: we map all horizontal and vertical boundaries of the image voxels and detector cells onto a common plane. We calculate the lengths of overlap along the x (or y) direction (a) and along the z direction (b) and we multiply them to get the area of overlap.

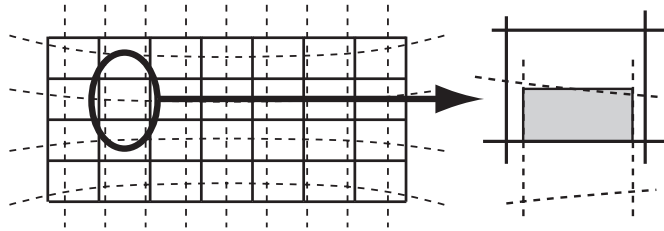


Figure 4. 3D distance-driven method: we map the voxels and detector cells onto a common plane in order to determine their projected area (solid lines and dashed curves) and approximate the area of overlap (grey rectangle).

2.3. Three dimensions

Perhaps the most straightforward way to extend the distance-driven method to 3D would be to apply the 2D version for all in-plane calculations (figure 3(a)) and combine it with interpolation in the z direction. However, this z interpolation is inefficient and results in artefacts, similar to the in-plane interpolation. Instead, we propose to extend the original concept behind the distance-driven technique to three dimensions. Every view now defines a bijection between the position on the 2D detector and the position within each image slab (parallel to the xz plane or to the yz plane). In other words, every point within an image slab is mapped uniquely onto a point on the detector, and vice versa. Thus we can define an *area* of overlap between each projected image voxel and each detector cell. We map the voxels and detector cells onto a common plane (e.g. the xz plane) in order to determine their projected area and the area of overlap (figures 3 and 4). In practice we only map the horizontal and vertical boundaries of the image voxels and detector cells, approximating their shape by a rectangle. We calculate the lengths of overlap along the x (or y) direction and along the z direction and then multiply them to get the area of overlap. This approach can be implemented as two nested loops containing the distance-driven kernel operation from equation (4). Note that a consequence of the area-of-overlap weighting is that the distance-driven technique is inherently $1/r^2$ -weighted.

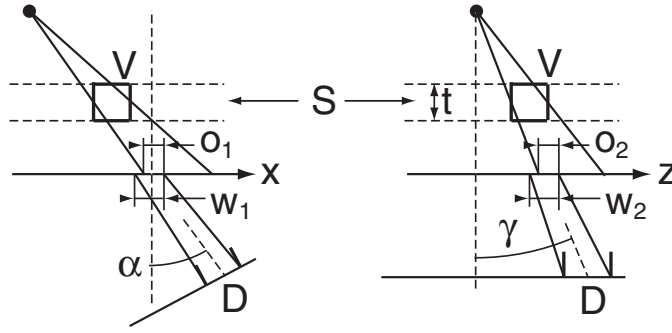


Figure 5. 3D distance-driven method: the contribution from a voxel V to the line integral of a detector D is weighted by the normalized lengths of overlap o_1/w_1 in-plane (left) and o_2/w_2 out-of-plane (right), as well as by the intersection length $t/(\cos \alpha \cos \gamma)$ of the line of interest with slab S .

Figure 5 gives a more detailed representation of the 3D distance-driven method. To calculate the contribution of voxel V with attenuation μ_V to the line integral p_D for detector D , we need to multiply three factors:

- (1) The intersection length of the line of interest with the image slab S , the latter being defined as the slab parallel to the xz plane and containing voxel V . This intersection length is given by $t/(\cos \alpha \cos \gamma)$, where t is the isotropic voxel size, and α and γ are the in- and out-of-plane angles of the line of interest with the y -axis.
- (2) The in-plane normalized overlap between voxel V and detector D , which is given by o_1/w_1 , where o_1 is the length of overlap and w_1 is the detector width.
- (3) The out-of-plane (normalized) overlap between voxel V and detector D , which is given by o_2/w_2 , where o_2 is the length of overlap and w_2 is the detector width.

The total contribution of voxel V to detector D is given by the update equation

$$p_D \leftarrow p_D + \frac{t}{\cos \alpha \cos \gamma} \frac{o_1}{w_1} \frac{o_2}{w_2} \mu_V.$$

3. Experimental results

We have performed a number of experiments to evaluate the benefits of the distance-driven method. Artefact and performance results for the 2D case were presented in De Man and Basu (2002). In section 3.1 we study the artefact behaviour in 3D, in section 3.2 we give performance results in 3D, and in section 3.3 we give a detailed analysis of image noise versus spatial resolution for 3D reconstruction using the well-known algorithm of Feldkamp, Davis and Kress (FDK) (Feldkamp *et al* 1984).

3.1. Artefacts

We performed a pixel-driven and distance-driven projection of a uniform slab centred at the xz plane. Figures 6(a) and (b) show the projection at 45° . The pixel-driven projection shows interpolation artefacts, both in the in-plane and z directions (figure 6(a)). The artefacts are entirely eliminated using the distance-driven method (figure 6(b)). We also performed a ray-driven and distance-driven backprojection of one uniform view at a view angle of 45° . Figures 6(c) and (d) show the centre image slab (parallel to the xz plane) in the resulting

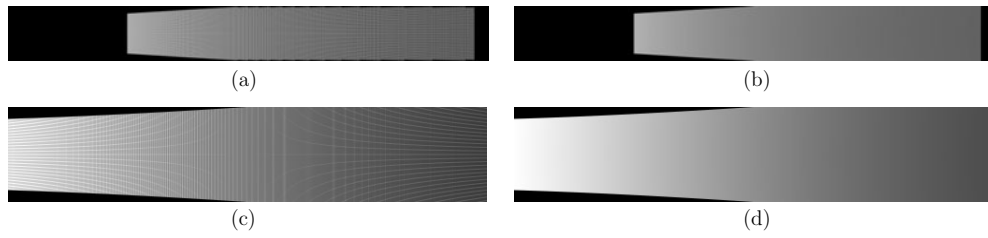


Figure 6. (a)–(b) 3D pixel-driven and distance-driven projection of a uniform slab, (c)–(d) 3D ray-driven and distance-driven backprojection of one uniform view.

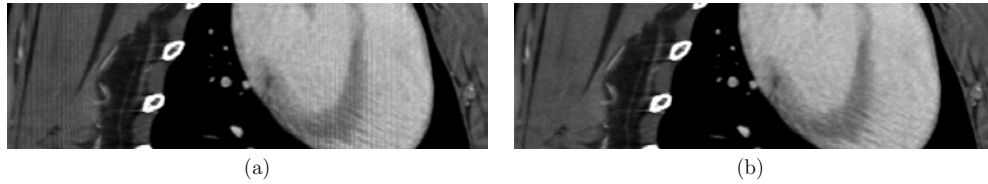


Figure 7. 3D iterative reconstruction of a CT scan of a rabbit heart: (a) ray-driven MLTR and (b) distance-driven MLTR.

volumes. The ray-driven backprojection shows interpolation artefacts, both in the x and in the z direction (figure 6(c)). The artefacts are entirely eliminated using the distance-driven method (figure 6(d)). These results show that the distance-driven technique avoids both image-domain and sinogram-domain high-frequency artefacts.

We also performed iterative MLTR (maximum likelihood for transmission, Nuyts *et al* (1998)) reconstructions (10 iterations of 40 subsets) of a rabbit thorax scan. The measurement was performed using a flat-panel-based volumetric CT prototype, with a detector cell size of $200\ \mu\text{m}$, a magnification of 1.5, and using 1024 detector columns and 370 detector rows. Figure 7 shows longitudinal reformats of the $512 \times 512 \times 160$ reconstruction. The ray-driven result shows some interpolation artefacts (vertical lines in figure 7(a)), which are entirely eliminated with the distance-driven method (figure 7(b)).

3.2. Computational performance

To assess computational performance, we performed distance-driven, ray-driven and pixel-driven backprojections for a curved cone-beam geometry and for different data sizes. The computation time for projection can be assumed to be comparable to backprojection. We performed experiments for both image size and sinogram size equalling n^3 , where n is the data size ranging between 15 and 250. The timing experiments were performed on a 3.06 GHz Pentium Xeon using the GNU 3.2 compiler suite. All backprojections were $1/r^2$ -weighted. The implementations used in this comparison were only moderately optimized. For instance we did not include approximations to the arc-tangent or exploit vectorization (SIMD parallelism). Despite the limitations of the current implementations, the comparison gives a good impression of the relative performance of the different methods. The results are shown in figure 8. The distance-driven technique yielded an eightfold improvement compared to the pixel-driven and a fivefold improvement compared to ray-driven methods. For comparison, in De Man and Basu (2002) we showed similar performance results for 2D fan-beam geometries.

The higher computational performance of the distance-driven technique is due to its low arithmetic complexity combined with a highly sequential memory access pattern.

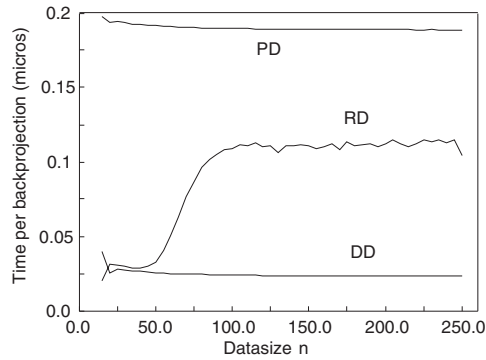


Figure 8. Computation time per backprojection (in micro-seconds) on a 3.06 GHz Pentium Xeon for pixel-driven (PD), ray-driven (RD) and distance-driven (DD) backprojection for different data sizes n . The time per backprojection is defined as the total computation time divided by n^4 .

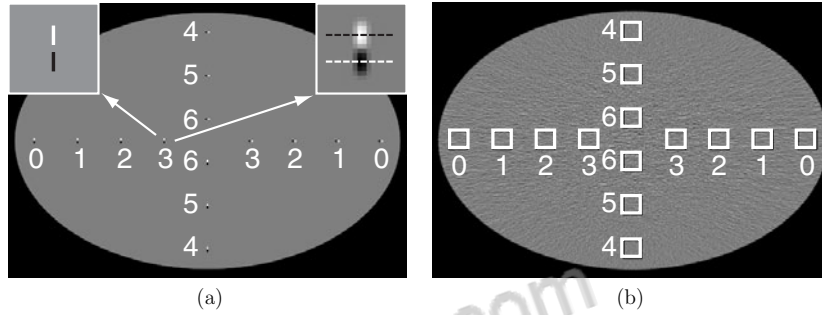


Figure 9. We simulated an elliptical water cylinder of $42 \text{ cm} \times 28 \text{ cm} \times 10 \text{ cm}$. For analysis of spatial resolution a number of thin lines were inserted (a). For analysis of image noise, Poisson noise was added to the simulated intensities prior to reconstruction (b).

Consequently, it is fast, well suited for hardware implementation, and it allows for good cache utilization on computers with hierarchical memories.

3.3. Noise and spatial resolution

Next, we compared the noise and resolution properties of 3D FDK reconstructions using pixel-driven and distance-driven backprojections. The overlap kernel of equation (4) allows for optimal smoothing based on the ratio between pixel size and detector cell size. Even if at isocentre, the image pixel size is perfectly matched to the effective detector aperture (including the effects of magnification and a possible smoothing kernel), the ratio between the two sizes changes as a function of position in the image. Pixel-driven backprojection does not exploit this, leading to poorer noise performance at the same resolution. To evaluate this effect, we analytically defined an elliptical water cylinder ($42 \text{ cm} \times 28 \text{ cm} \times 10 \text{ cm}$) and simulated a curved detector cone-beam geometry with 550×32 detector channels, 984 views, a magnification of 1.75, a $1 \text{ mm} \times 8 \text{ mm}$ focal spot size with a 0° target angle, and a 1.47 mm^2 detector cell size.

For analysis of the spatial resolution we inserted a number of horizontal and vertical thin lines (left inset in figure 9(a)) at different (x, y) -locations and at 3 mm from the mid-plane. We combined lines with positive and negative contrast to reduce the nonlinear partial volume

Table 1. Comparison of resolution (FWHM (mm)) and noise (RMS (cm^{-1})) for different ROIs for a $512 \times 512 \times 16$ FDK reconstruction.

ROI	FWHM _{PD}	FWHM _{DD}	$\Delta\%$	$\sigma_{\text{PD}}^{\text{adj}}$	$\sigma_{\text{DD}}^{\text{adj}}$	$\Delta\%$
0	1.134	1.172	3.3	0.0309	0.0249	-19.5
1	1.029	1.051	2.2	0.0350	0.0323	-7.7
2	0.996	1.007	1.1	0.0350	0.0344	-1.8
3	1.021	1.014	-0.7	0.0315	0.0315	-0.1
4	1.059	1.060	0.1	0.0167	0.0168	0.0
5	1.011	1.022	1.1	0.0247	0.0257	4.1
6	0.905	0.898	-0.7	0.0391	0.0396	1.2

effect. The lines are 0.1 mm thick and the contrast is $\pm 0.128 \text{ cm}^{-1}$. The reconstructed image in figure 9(a) illustrates the different line locations. The radial, azimuthal and longitudinal resolutions were determined separately by least-squares fitting of a Gaussian function to the line spread function (right inset in figure 9(a)) and then computing the full-width-at-half-maximum (FWHM) from the fit.

For analysis of the image noise (figure 9(b)) we added Poisson noise to a simulation of the same elliptical water cylinder (now without the lines), based on an air scan intensity of 10^6 photons per view per detector cell. We measured the RMS error due to noise in different regions in the image (two series numbered from 0 to 6). The final noise measures were obtained by taking the arithmetic average of each pair of symmetrically positioned regions.

It is important to understand the trade-off between noise and resolution. The resolution for any technique can freely be modulated (up to a certain level) by using boosting or smoothing kernels. However, the noise level at a given resolution reflects the dose efficiency, which is a more fundamental property of a given technique.

We performed standard FDK reconstructions on a $512 \times 512 \times 16$ reconstruction grid. The voxel size was 0.84 mm isotropically, which matches the detector cell size as seen at isocentre. For each region-of-interest (ROI), we evaluated the radial, azimuthal, and longitudinal spatial resolutions using both techniques, and we calculated the geometric average (FWHM_{PD}, FWHM_{DD}) and the percent difference ($\Delta\%$). We also evaluated the noise for both techniques, and we calculated the percent improvement of the noise level for equal resolution ($\Delta\%$). The resolutions were first matched in each of the three directions on a per-ROI basis by applying a Gaussian smoothing with the appropriate sigma to the image with the best spatial resolution. The noise level was evaluated after this smoothing procedure, resulting in *adjusted* sigmas, which we denote by the superscript *adj* ($\sigma_{\text{PD}}^{\text{adj}}$, $\sigma_{\text{DD}}^{\text{adj}}$). The results are shown in table 1. For this particular pixel size, the distance-driven technique has a slightly decreased (3%) native spatial resolution for the most peripheral regions. After matching the spatial resolution, the distance-driven method results in a noise reduction of up to 20% compared to pixel-driven. To verify that the noise results are not affected by image artefacts we also computed the sigma in a noise-free simulation. Indeed, the sigma without noise was three to four orders of magnitude smaller than in the case with noise. All results were obtained without a bow-tie filter. There may be a reduction in relative performance when a bow-tie filter is included.

4. Conclusions

In this paper, we have extended the distance-driven technique for projection and backprojection to three dimensions. We have shown that the distance-driven technique eliminates the artefacts

seen in ray-driven backprojection and in pixel-driven projection. We have performed 3D iterative reconstruction of a volumetric CT scan of a rabbit thorax. The results demonstrate a reduction in artefacts for distance-driven iterative reconstruction compared to ray-driven iterative reconstruction. In terms of computation time, the distance-driven method combines a sequential memory access pattern (similar to pixel-driven methods) with a low arithmetic cost (similar to ray-driven methods). This combination results in a significant reduction in computation time. We have evaluated the performance benefit for 3D geometries, and the results show an eightfold improvement compared to the pixel-driven method and a fivefold improvement compared to the ray-driven method. We have performed 3D distance-driven and pixel-driven FDK reconstructions of software phantom simulations to show the benefit of the distance-driven method for FBP reconstruction in terms of image noise and spatial resolution. Under the described conditions, we achieved up to a 20% noise reduction at matched spatial resolution. In addition to its image quality and performance benefits, the distance-driven method is well suited for hardware implementation and flexible in terms of allowable imaging geometries.

Acknowledgments

We wish to thank Walt Dixon for his help with the performance optimization of the different distance-driven implementations, and Pete Edic, Bob Senzig and Jean-Baptiste Thibault for their valuable feedback on the manuscript.

References

- Basu S 2000 Fast algorithms for tomography *PhD Thesis* University of Illinois at Urbana-Champaign
- Basu S and Bresler Y 2000 An $O(N^2 \log N)$ filtered backprojection reconstruction algorithm for tomography *IEEE Trans. Image Process.* **9** 1760–73
- Buonocore M, Brody W and Macovski A 1981 A natural pixel decomposition for two-dimensional image reconstruction *IEEE Trans. Biomed. Eng.* **28** 69–78
- Cho Z, Chen C and Lee S 1990 Incremental algorithm—a new fast backprojection scheme for parallel beam geometries *IEEE Trans. Med. Imaging* **9** 207–17
- De Man B 2001 Iterative reconstruction for reduction of metal artifacts in computed tomography *PhD Thesis* University of Leuven (http://bilbo.esat.kuleuven.ac.be/web-pages/mic_papers/Bruno.DeMan/PHD.pdf)
- De Man B and Basu S 2002 Distance-driven projection and backprojection *IEEE Nuclear Science Symp. Medical Imaging Conf. (Norfolk)*
- De Man B and Basu S 2003 3D distance-driven projection and backprojection *Proc. 7th Int. Conf. on Fully 3D Reconstruction in Radiology and Nuclear Medicine (Saint Malo)*
- De Man B and Basu S 2004 Distance-driven projection and backprojection *IEEE Trans. Med. Imaging* submitted
- De Man B, Nuyts J, Dupont P, Marchal G and Suetens P 1999 Metal streak artifacts in X-ray computed tomography: a simulation study *IEEE Trans. Nucl. Sci.* **46** 691–6
- De Man B, Nuyts J, Dupont P, Marchal G and Suetens P 2001 An iterative maximum-likelihood polychromatic algorithm for CT *IEEE Trans. Med. Imaging* **20** 999–1008
- Feldkamp L, Davis L and Kress J 1984 Practical cone-beam algorithm *J. Opt. Soc. Am. A* **1** 612–9
- Fessler J 1994 Penalized weighted least-squares image reconstruction for positron emission tomography *IEEE Trans. Med. Imaging* **13** 290–300
- Glick S and Soares E 1998 Noise characteristics of SPECT iterative reconstruction with a mis-matched projector-backprojector pair *IEEE Trans. Nucl. Sci.* **45** 2183–8
- Herman G 1980 *Image Reconstruction from Projections* (Orlando: Academic) p 63
- Hsieh J, Molthen R, Dawson C and Johnson R 2000 An iterative approach to the beam hardening correction in cone beam CT *Med. Phys.* **27** 23–9
- Hsieh Y, Gullberg G, Zeng G and Huesman R 1996 Image reconstruction using a generalized natural pixel basis *IEEE Trans. Nucl. Sci.* **43** 2306–19
- Joseph P 1983 An improved algorithm for reprojecting rays through pixel images *IEEE Trans. Med. Imaging* **1** 192–6

- Kotian F, Martin G, Metz S, Collins A and Wu M 1995 Backprojection for X-ray CT system *United States Patent* US5473654
- Lacroute P and Levoy M 1994 Fast volume rendering using a shear-warp factorization of the viewing transformation *Proc. SIGGRAPH 1994 (Orlando)* pp 451–8
- Lewitt R 1992 Alternatives to voxels for image representation in iterative reconstruction algorithms *Phys. Med. Biol.* **37** 705–16
- Mueller K, Yagel R and Wheller J 1999 Fast implementations of algebraic methods for three-dimensional reconstruction from cone-beam data *IEEE Trans. Med. Imaging* **18** 538–48
- Nuyts J, De Man B, Dupont P, Defrise M, Suetens P and Mortelmans L 1998 Iterative reconstruction for helical CT: a simulation study *Phys. Med. Biol.* **43** 729–37
- Peters T 1981 Algorithms for fast back- and re-projection in computed tomography *IEEE Trans. Nucl. Sci.* **28** 3641–7
- Siddon R 1985 Fast calculation of the exact radiological path length for a three-dimensional CT array *Med. Phys.* **12** 252–5
- Suetens P 2002 *Fundamentals of Medical Imaging* (Cambridge: Cambridge University Press)
- Yu D and Huang S 1993 Study of reprojection methods in terms of their resolution loss and sampling errors *IEEE Trans. Nucl. Sci.* **40** 1174–8
- Zeng G and Gullberg G 1993 A ray-driven backprojector for backprojection filtering and filtered backprojection algorithms *IEEE Nuclear Science Symp. Medical Imaging Conf. (San Francisco)* pp 1199–201
- Zeng G and Gullberg G 2000 Unmatched projector/backprojector pairs in an iterative reconstruction algorithm *IEEE Trans. Med. Imaging* **19** 548–55
- Zhuang W, Gopal S and Hebert T 1994 Numerical evaluation of methods for computing tomographic projections *IEEE Trans. Nucl. Sci.* **41** 1660–5
- Zwicker M, Pfister H, van Baar J and Gross M 2002 EWA splatting *IEEE Trans. Vis. Comp. Graph.* **8** 223–38



知网查重限时 7折 最高可优惠 120元

本科定稿，硕博定稿，查重结果与学校一致

立即检测

免费论文查重: <http://www.paperyy.com>

3亿免费文献下载: <http://www.ixueshu.com>

超值论文自动降重: http://www.paperyy.com/reduce_repetition

PPT免费模版下载: <http://ppt.ixueshu.com>
