

High-Quality Volumetric Reconstruction on Optimal Lattices for Computed Tomography

Bernhard Finkbeiner¹, Usman R. Alim¹, Dimitri Van De Ville², and Torsten Möller¹

¹Simon Fraser University, Canada

²University of Geneva, Switzerland

Abstract

Within the context of emission tomography, we study volumetric reconstruction methods based on the Expectation Maximization (EM) algorithm. We show, for the first time, the equivalence of the standard implementation of the EM-based reconstruction with an implementation based on hardware-accelerated volume rendering for nearest-neighbor (NN) interpolation. This equivalence suggests that higher-order kernels should be used with caution and do not necessarily lead to better performance. We also show that the EM algorithm can easily be adapted for different lattices, the body-centered cubic (BCC) one in particular. For validation purposes, we use the 3D version of the Shepp-Logan synthetic phantom, for which we derive closed-form analytical expressions of the projection data. The experimental results show the theoretically-predicted optimality of NN interpolation in combination with the EM algorithm, for both the noiseless and the noisy case. Moreover, reconstruction on the BCC lattice leads to superior accuracy, more compact data representation, and better noise reduction compared to the Cartesian one. Finally, we show the usefulness of the proposed method for optical projection tomography of a mouse embryo.

Categories and Subject Descriptors (according to ACM CCS): Image Processing and Computer Vision [I.4.5]: Reconstruction Transform methods—

1. Introduction

3D Computed Tomography (CT) provides a way to reconstruct volumetric data from a set of 2D projections of the 3D object acquired at various projection angles. The Maximum Likelihood Expectation Maximization (EM) algorithm is a widely used CT method for image reconstruction in emission tomography (ET). The EM algorithm is an iterative method that consists of three steps in each iteration: forward projection, correction, and back-projection. The forward projection can be conveniently implemented as volume rendering that enables an implementation of the EM algorithm on commodity graphics hardware that delivers very high performance at low hardware costs [CM03, Xu07].

Volume rendering usually seeks to reconstruct a continuous-domain representation from discrete samples stored on a regular lattice. Reconstruction is achieved by convolving the discrete samples with a continuous reconstruction kernel. Obviously, the quality of the reconstruction depends on the chosen reconstruction kernel. In general,

higher-order kernels that use a larger neighborhood deliver better quality than lower-order ones. This intuitive fact is usually adapted by using trilinear interpolation when implementing the forward projection in the EM algorithm within a hardware-accelerated volume rendering framework [CM03].

In Section 3 of this paper, we will show that a volume rendering implementation of the EM algorithm is correct if and only if it uses nearest-neighbour (NN) interpolation. Higher-order reconstruction kernels are not compatible with standard EM. To the best of our knowledge this is a novel result.

We substantiate this result empirically with several tests in Section 5 demonstrating that NN interpolation achieves most accurate results compared to higher-order filters.

To accelerate the algorithm, we implement the EM algorithm using commodity graphics hardware. Since the EM algorithm is not bound to a specific lattice, we extend the implementation to the BCC lattice that is known to have better sampling properties than the Cartesian (CC) lattice. This enables reconstruction of volumetric data directly on

a BCC lattice without the need to change acquisition devices. We therefore pave the way to a more wide-spread production of data sampled on the BCC lattice: By using a BCC lattice instead of a CC lattice, either a reduction of 29% of the samples without any loss of information can be achieved [TMG01], or more detail can be captured for the same number of samples.

The three major contributions of our work are as follows:

First, we implement the EM algorithm using volume rendering and show in Section 3 that the traditional discrete representation of the EM algorithm is replaced by a continuous-domain one using a volume rendering framework. We show that this requires the use of a NN filter.

Second, in Section 5 we show that volumetric data reconstructed on a BCC lattice deliver higher accuracy and better noise reduction compared to the CC lattice.

Third, our GPU implementation is two orders of magnitude faster than a reference CPU implementation: large volumes with approximately 130 million samples are reconstructed within less than an hour compared to several days.

We test our method using an analytical phantom and a real-world dataset; i.e., we use the 3D version of the well known Shepp-Logan (SL) phantom [SL74] and we derive analytical expressions for the ideal 2D projection data. Furthermore, we employ data from a new modality where projections of a mouse embryo were acquired using optical projection tomography (OPT) [Sha04].

2. Related Work

Algorithms for CT reconstruction can be split into two categories: analytical and iterative methods. Analytical methods invert the Radon transform in a single step and include back-projection methods such as Filtered Back Projection (FBP) [Dea83, FDK84]. They only work reliably if the noise is limited; e.g., for transmission tomography. Iterative methods, on the other hand, optimize a criterion to reconstruct the volume; e.g., the Algebraic Reconstruction Technique (ART) [GBH70] or the EM algorithm [SV82] alternate between forward and back-projections. Iterative procedures are computationally expensive, but the criterion can be chosen to improve robustness in the presence of noise [LM03], which is clearly the case for ET.

There are several CT methods that have been successfully ported to the GPU improving reconstruction times [CM03, XM07b]. However, all these methods work on the standard CC lattice and/or suffer from the poor resolution of 8 bits of the earlier GPU's framebuffer. An excellent overview can be found in Sitek's tutorial [Sit07] and in Xu's thesis [Xu07]. None of these methods have investigated the influence of different filter kernels (and mostly used linear interpolation). In contrast, we show that highest quality is achieved when using a NN filter.

The use of BCC lattices in tomography research is not new [ML95, MY96]. Matej et al. [ML95] used spherically-symmetric volume elements (blobs) introducing computational overhead due to the overlap. Mueller et al. [MY96] employed the BCC lattice to reduce the computational costs to a factor of 70.5% of the equivalent CC lattice. More recent work by Xu et al. [XM07a] uses commodity graphics hardware to accelerate the FBP algorithm using BCC lattices. As mentioned above, this method is not reliable when reconstructing from noisy projections. Iterative methods that can deal better with noise often require a forward projection step in every iteration. Forward projection, however, involves rendering of a volume; i.e., it involves reconstruction from the volume on the underlying lattice with a kernel.

Since we use the EM algorithm (which is an iterative method) in a rendering framework, the forward projection is basically achieved via volume rendering. Volume rendering involves reconstruction of a continuous-domain function from discrete samples lying on a regular lattice. Theußl et al. [TMG01] were the first to use BCC lattices together with a spherical extension of reconstruction filters in the area of volume rendering. However, visual results were not convincing and since then several filters for the BCC lattice have been proposed: box splines [EDM04, EVM08], a prefiltering operator followed by a Gaussian filter [Csé05] and a B-spline filter [CH06], and BCC-splines [Csé08]. In this paper we choose box splines since they guarantee approximation order and are numerically stable. On the CC side we employ the commonly used B-splines.

3. Computed Tomography

We present a brief overview of the EM algorithm, first in its traditional discrete form as introduced by Shepp et al. [SV82] and then in a continuous form that can be adapted for use in a volume rendering framework.

3.1. EM Reconstruction

The goal of image reconstruction in ET is to estimate a continuous 3D activity distribution from discrete measurements that correspond to some integral transformation of the activity distribution. Lewitt et al. [LM03] have classified the models used to represent the data collection process into three major categories, namely discrete-continuous (D-C), discrete-discrete (D-D) and continuous-continuous (C-C). D-C models are a natural setting for ET and attempt to relate the discrete measurements to the continuous 3D activity distribution. D-D models are obtained from D-C models by using a finite set of basis functions to represent the unknown activity distribution. A reconstruction algorithm is then used to estimate the coefficients of the basis functions. C-C models interpret the discrete measurements as samples of a continuous function in the measurement space and an analytic formula is used to invert the integral transform in order to estimate the activity distribution.

Here, we shall focus on the EM algorithm which is based on a D-D model of data collection; i.e., it treats the acquired projections and the reconstruction volume as discrete data. The 3D volume is typically discretized into non-overlapping cubic cells that are tiled such that their centers make up a regular 3D CC lattice. However, the algorithm is not tied to any specific grid and allows the use of non-CC lattices.

We denote by $\lambda(x)$, the unknown 3D activity distribution. For convenience, we lexicographically order the cells and index them with j ($j = 1, \dots, J$). Similarly, assuming a total of I detector bins, we order them lexicographically and index them with i ($i = 1, \dots, I$). We denote the total number of photons detected in bin i as y_i and the total activity in cell j as λ_j . The stochastic nature of photon detections in a particular bin is modeled as a Poisson process that is independent of the photon arrivals in other bins. In particular, photon arrivals in bin i follow a Poisson distribution with mean $\sum_j a_{i,j} \lambda_j$, where the term $a_{i,j}$ models the physics and geometry of the imaging process and represents the probability that a photon emitted anywhere in cell j will be detected in bin i . This allows the D-D imaging model to be written as a matrix-vector product given by

$$E[\hat{\mathbf{y}}] = \mathbf{A} \cdot \hat{\boldsymbol{\lambda}}, \quad (1)$$

where $\hat{\mathbf{y}}$ is the column vector $(\hat{y}_1, \dots, \hat{y}_I)^T$ consisting of detector counts and $\hat{\boldsymbol{\lambda}}$ is the column vector $(\hat{\lambda}_1, \dots, \hat{\lambda}_J)^T$ consisting of cell activities.

The EM algorithm takes the form of an iterative procedure that finds the image estimate $\hat{\boldsymbol{\lambda}}$ that maximizes the likelihood of measuring the data $\hat{\mathbf{y}}$ under the imaging model (1). We refer the reader to [SV82, LC84] for details of the derivation. If we denote an activity estimate at iteration n as $\hat{\boldsymbol{\lambda}}^{(n)}$, then the equation that updates the activity estimate of cell j , can be written as

$$\hat{\lambda}_j^{(n+1)} = \frac{\hat{\lambda}_j^{(n)}}{\sum_i a_{i,j}} \sum_i a_{i,j} \frac{y_i}{\sum_k a_{i,k} \hat{\lambda}_k^{(n)}}. \quad (2)$$

Let $\hat{\mathbf{p}}^{(n)}$ be the column vector $(\hat{p}_1^{(n)}, \dots, \hat{p}_I^{(n)})^T$ that represents the projection of the activity estimate $\hat{\boldsymbol{\lambda}}^{(n)}$ at iteration n . It is given by

$$\hat{\mathbf{p}}^{(n)} = \mathbf{A} \cdot \hat{\boldsymbol{\lambda}}^{(n)}. \quad (3)$$

Also, let $\hat{\mathbf{e}}^{(n)}$ be the column vector $(\hat{e}_1^{(n)}, \dots, \hat{e}_I^{(n)})^T$ consisting of correction factors. Equation (2) can then be expressed in a more concise form as

$$\hat{\lambda}_j^{(n+1)} = \hat{\lambda}_j^{(n)} \hat{u}_j^{(n)}, \quad (4)$$

where $\hat{u}_j^{(n)} = \frac{(\mathbf{A}^T \hat{\mathbf{e}}^{(n)})_j}{\sum_i a_{i,j}}$ and \mathbf{A}^T is the transpose of \mathbf{A} .

The update equations (2) and (4) have a simple interpretation in terms of projection and back-projection operations. In the standard EM framework, the matrix \mathbf{A} computes a projection of the current activity estimate whereas its transpose, \mathbf{A}^T , back-projects correction factors into the volume to

update the estimate. Sometimes, a different back-projection matrix \mathbf{B}^T may be used to accelerate convergence, resulting in a modified reconstruction algorithm that goes by the name of Dual-Matrix reconstruction. In ET, it accelerates convergence by modeling the physics (attenuation, Compton scattering, detector blurring) as well as the geometry of the acquisition process into the projection matrix \mathbf{A} while keeping the back-projection matrix \mathbf{B}^T sparse (e.g by modeling geometry only). The projection and back-projection matrices must form a valid pair as analyzed in [ZG00].

3.2. Volume Rendering Formulation

Volume rendering techniques for tomographic reconstruction have the advantage that they avoid the expensive computation and storage of the systems matrix \mathbf{A} , rather, the entries of the matrix are implicitly computed on the fly during the projection and back-projection steps. In order to make use of volume rendering techniques, we need to write the discrete quantities presented in the previous section, in a form that is more suitable for volume rendering algorithms. In particular, the activity distribution $\lambda(x)$ and the forward projection model (1) need to be transposed into the continuous domain. Towards this end, we formulate the equations in terms of arbitrary basis functions and highlight the conditions under which the continuous-domain representation is equivalent to the discrete EM framework.

We start by examining the D-C data collection model which treats both the emission density and the detection probability as continuously defined functions. Let $h_i(x)$ denote the continuous-domain probability density function that represents the probability that an emission within an infinitesimal volume at x will be detected in bin i . If we represent the continuous-domain emission density as a linear combination of a finite number of basis functions, the continuous analogue of the discrete imaging model (1) can be written as

$$\hat{g}_i = \int_{\mathbb{R}^3} h_i(x) \left(\sum_j \lambda_j \phi_j(x) \right) dx, \quad (5)$$

where \hat{g}_i denotes the total number of photons detected in bin i under a continuous imaging model, $\phi_j(x)$ is the basis function corresponding to cell j and λ_j is the corresponding coefficient. Here, we have not placed any restrictions on the choice of basis functions. Any set of functions that form a valid basis can be used.

An entry of the matrix \mathbf{A} represents the average probability that an emission from cell j will be detected in bin i [LM03]. It can be obtained through $h_i(x)$ by integrating over the volume occupied by cell j ; i.e.,

$$a_{i,j} = \int_{\mathbb{R}^3} h_i(x) \chi_j(x) dx, \quad (6)$$

where $\chi_j(x)$ denotes the characteristic function of cell j . It is unity within the volume of the cell and zero elsewhere.

Using (6) and the D-D projection model of the EM framework (1), we can write the expected total number of photons

detected in bin i as

$$\begin{aligned} \sum_j \hat{\lambda}_j a_{i,j} &= \sum_j \hat{\lambda}_j \left(\int_{\mathbb{R}^3} h_i(x) \chi_j(x) dx \right) \\ &= \int_{\mathbb{R}^3} h_i(x) \left(\sum_j \hat{\lambda}_j \chi_j(x) \right) dx. \end{aligned} \quad (7)$$

By comparing this with the continuous imaging model (5), we observe that if we use the characteristic functions $\chi_j(x)$ as basis functions along with the total cell emissions $\hat{\lambda}_j$ as the corresponding coefficients, the continuous imaging model becomes equivalent to the discrete formulation of the EM framework. Thus, a continuous imaging model that makes use of NN interpolation for the emission density and treats the integration kernel $h_i(x)$ as a continuously defined function, can safely be used as a projector in an EM reconstruction framework. Higher-order interpolation schemes are incompatible. This should come as no surprise since the EM formulation capitalizes on the additive property of independent Poisson distributions. A natural way to achieve this independence is to assume an underlying piecewise-constant emission density.

The back-projection step can also be expressed in an integral form similar to (7). Taking (6) and substituting it into (2), we see that the total unnormalized correction factor back-projected to cell j is given by

$$\sum_i \hat{c}_i a_{i,j} = \sum_i \left(\int_{\mathbb{R}^3} \hat{c}_i h_i(x) \chi_j(x) dx \right). \quad (8)$$

Equations (7) and (8) are in a more suitable form since volume rendering algorithms are fine tuned for the purpose of computing such integrals.

In ET modalities, a collimator is usually employed that only allows photons traveling perpendicular to the detector plane to pass through. If we further neglect the effects of photon scattering, the integration kernel $h_i(x)$ is non-zero within a cuboid-shaped beam perpendicular to the detector plane and zero outside. Forward projection (7) can therefore be efficiently computed via volume rendering using orthographic projection and an emission-only integrator that samples the volume along rays perpendicular to the image plane. Similarly, the back-projection (8) can be evaluated by treating each image plane of correction factors as a light source and tracing rays through the volume, accumulating their contribution at each cell. In such a volume rendering based approach for computing the forward and back-projection, if the integrands are sampled at the same locations, we have a matched projector/back-projector pair as in (4). On the other hand, in hardware-accelerated reconstruction, the usual approach is to use a smaller step size for the forward projection and an approximate evaluation of the back-projection step, thus leading to an unmatched approach.

As mentioned earlier, this formulation can be generalized to any 3D lattice. Once the lattice is determined, the only adjustment needed is in the choice of the characteristic function $\chi_j(x)$, which changes according to the lattice's Voronoi

cell. In our case we choose the CC and BCC lattices which have the cube and the truncated octahedron as their Voronoi cells respectively. Therefore, $\chi_j(x)$ is the NN interpolation kernel of the lattice.

4. Hardware-Based Implementation

The EM algorithm expects m projections (\hat{y}) as input that are equidistantly distributed over a certain range (usually 360 degrees). The algorithm consists of three different steps in each iteration n : Forward projection, correction step, and back-projection (see Figure 1).

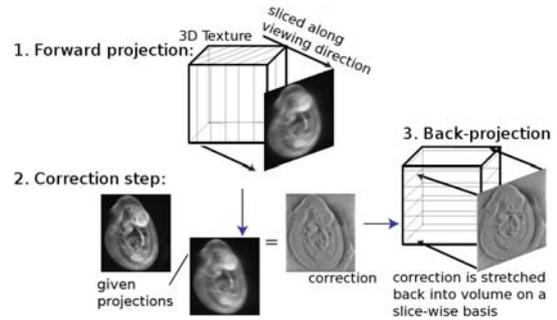


Figure 1: Overview of the EM implementation. Illustration adapted by permission [BDCM05].

Forward Projection: For each iteration n , the forward projection computes m projection estimates $\hat{\mathbf{p}}^{(n)}$ from the current 3D estimate $\lambda(x)^{(n)}$ (where $\lambda(x)^{(1)} = 1$). Note that Equation 7 can be considered as beam integrals that can be approximated by line integrals that are easily computed via volume rendering. Therefore, we render $\lambda(x)^{(n)}$ from every projection angle to obtain $\hat{\mathbf{p}}^{(n)}$. According to Equation 7, a strict implementation relies on NN interpolation. Here, we compare NN interpolation to higher-order kernels to empirically confirm the theoretical results from Section 3.

To conduct GPU-based volume rendering on the CC and BCC lattice, two different adjustments have to be made: To account for different filters, the fragment shaders have to be adjusted (see Section 5.2) and for the BCC lattice we need a different memory access scheme: Figure 2 (a) shows a BCC lattice that can be seen as two interleaved CC lattices (red and blue) where the samples of the second lattice are shifted to the center of the first lattice's cubes. A BCC lattice point (x, y, z) has either only even values x, y and z , or they are all odd. By storing the BCC lattice in a 3D array and using the following mapping, a fast conversion of a BCC point (x, y, z) to its index (i, j, k) in the 3D array is achieved by $i = x \div 2$, $j = y \div 2$ and $k = z$ where \div is integer division. Being able to store the two interleaved CC lattices as one 3D array, it is loaded as a 3D texture into GPU memory (Figure 2 (b, c)).

NN search for a point p in a BCC lattice is achieved by finding the two closest points to p in the two CC lattices that

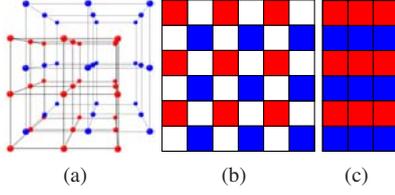


Figure 2: (a): A BCC lattice is formed by two CC lattices (red and blue). (b): 2D scheme of the BCC lattice. (c): In memory, every odd (blue) row (i.e., slice in 3D) is shifted to the center of the "even" (red) CC lattice.

form the BCC lattice. Having found these two points, one chooses the closest point to p .

We use texture slicing with an orthographic camera to render the estimate of the volume. No attenuation correction takes place so we use additive alpha blending. To compute the line integral per pixel, we multiply the sum of the fragments with the distance between two adjacent slices (i.e., the stepsize). This results in images $\hat{\mathbf{p}}^{(n)}$ similar to X-ray projections.

Correction Step: The correction images are calculated according to Equation 4; i.e., we compute the correction vector $\hat{\mathbf{c}}^{(n)} = (\hat{y}_1/\hat{p}_1^{(n)}, \dots, \hat{y}_l/\hat{p}_l^{(n)})^T$.

Back-Projection: For the back-projection we compute the integral according to Equation 8. To compute the next estimate $\lambda(x)^{(n+1)}$, $\lambda(x)^{(n)}$ conceptually becomes the camera (i.e., voxels detect photons) and $\hat{\mathbf{c}}^{(n)}$ acts as light source (i.e., pixels emit photons). We proceed as proposed by Chidlow et al. [CM03]: For every horizontal slice of $\lambda(x)^{(n)}$, we find the corresponding row of pixels in $\hat{\mathbf{c}}^{(n)}$. We load this row as a 1D texture into GPU memory which is then "smeared" over a (rotated) quadratic polygon which represents a horizontal slice in the *update volume* $\hat{\mathbf{u}}^{(n)}$. For every rotation angle of the m projections, these values are normalized by m and get accumulated in the rendering buffer. This polygon represents one slice of $\hat{\mathbf{u}}^{(n)}$. Finally, we obtain the next estimate $\lambda(x)^{(n+1)} = (\hat{u}_1^{(n)} \hat{\lambda}_1^{(n)}, \dots, \hat{u}_j^{(n)} \hat{\lambda}_j^{(n)})^T$.

CC vs. BCC: Differences in the algorithm when using the BCC lattice only occur in the forward projection (where the fragment shaders and the storage scheme for the BCC lattice have to be adjusted), and in the back-projection: A BCC lattice is formed by two CC lattices. Therefore, in the BCC lattice every slice in the xy -, xz - and yz -plane is a 2D CC lattice where every odd slice is shifted by one half of the grid spacing. This shift has to be taken into account for every odd polygon during back-projection.

5. Results

In this section we test our GPU implementation of the EM algorithm. We ran our experiments on a PC with a Nvidia GeForce 9800 GX2 graphics board using Linux 2.6.18 with

gcc 4.1.2, OpenGL, and GLSL. In Section 5.1, we derive the 3D Shepp-Logan (SL) phantom with analytically computed projections. In Section 5.2, we investigate the influence of different filters in the forward projection step, and Section 5.3 demonstrates the superiority of the BCC lattice over the CC lattice. Finally, Section 5.4 demonstrates the feasibility of our method for a real-world dataset.

5.1. Synthetic Phantom

In practical CT settings only the projections are given and a ground truth is not available. In order to be able to compare CT algorithms we have to measure the error between the result (the estimate of the volume) and the actual solution (i.e., a ground truth of the volume). Hence, it is desirable to define synthetic phantoms and to create (ideal) projections for which we can compute an error measure. Therefore, we employ the SL dataset. The 2D SL test function was introduced by Shepp and Logan [SL74]. It consists of several analytically defined ellipsoids that resemble the shape and characteristics of a slice of a brain's CT scan. We make use of its 3D version [KS88] and use it as ground truth.

We created 256 projections of the 3D SL phantom of size 128×128 each on a 360 degree orbit around the phantom. In order to obtain accurate projections, it is not desirable to compute the projections in a numerical way (i.e., by computing numerical integrals along rays cast through the volume onto the detector plane). Since the 3D SL phantom is a sum of ellipsoids, we can create the projections analytically by using the *Fourier slice theorem*: The Fourier transform of a unit solid sphere centered at the origin is given by [Miz73]

$$\hat{f}_s(\omega) := \frac{J_{3/2}(2\pi\|\omega\|)}{\|\omega\|^{3/2}}, \quad (9)$$

where $J_{3/2}$ is the Bessel function of the first kind of order $3/2$. Furthermore, an ellipsoid is obtained by an affine transformation of a sphere. These transforms can also be applied in the Fourier domain to yield the Fourier transform of an ellipsoid (and thus of the 3D SL phantom).

$$\hat{f}_e(\omega) := |\mathbf{M}| \exp(-2\pi i x_o^T \omega) \hat{f}_s(\mathbf{M}^T \omega), \quad (10)$$

where \mathbf{M} is an affine transformation matrix that maps the unit sphere to the ellipsoid and x_o is the center of the ellipsoid. Now a spatial projection can be created by first sampling the Fourier transform of the 3D SL phantom on a discrete 2D slice perpendicular to the viewing direction, centered at the origin, and then applying the inverse Fourier transform on this slice.

In real-world applications projections are often corrupted by noise. To test the denoising capabilities of the EM algorithm, we created two more sets of projections that were corrupted by Poisson noise: For each pixel p_{xy} in projection p , p_{xy} is treated as the mean of the Poisson distribution at pixel p_{xy} . A scaling factor n is used to control the mean. The higher n , the higher is the noise level and the lower is the

peak signal-to-noise ratio (PSNR). The first set has a PSNR of 32.19 and the second set has a PSNR of 22.19 (Figure 3).

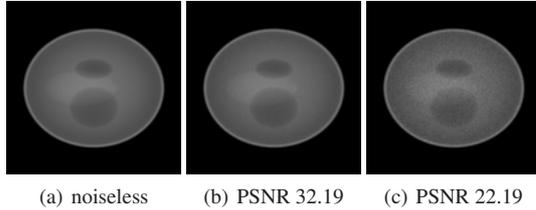


Figure 3: Analytically computed projections with different noise levels of the 3D SL phantom.

Having at hand the ground truth and the projections, the reconstructions are computed for 30 iterations while monitoring the root mean squared error (RMSE) between each estimate and the ground truth. We reconstructed the SL phantom on a CC lattice of size $128 \times 128 \times 128$ and on a BCC lattice of size $100 \times 100 \times 200$. The CC lattice has a slightly higher sampling density than the BCC lattice.

5.2. Different Filters

In Section 3 we have shown the necessity of using a NN filter. To substantiate this result empirically, we compare NN interpolation to higher-order reconstruction filters in the forward projection. For higher-order filtering on the BCC lattice we used the linear box spline and the quintic box spline with and without prefiltering (see [FEMV09] for details). For the CC lattice we used the trilinear B-spline and the tricubic B-spline with and without prefiltering. The linear box spline and the trilinear B-spline guarantee C^0 continuity, the quintic box spline and the tricubic B-spline C^2 continuity.

Figure 5 shows the RMSE after iteration 30 for each lattice, and the four respective kernels when using the noiseless and the two noisy projection sets. We recognize that NN interpolation (compared to higher-order reconstruction) delivers the most accurate result for both lattices, especially in the presence of noise.

In our model, we assume that only photons traveling perpendicularly to the sensor are recorded, and that distributions are independent. This should favor NN interpolation when reconstructing small features. Thus, we used the method described in Section 5.1 to create 256 noiseless projections (40×40) of small spheres with radius $\approx 2x$ (where x is the side length of a cubic voxel), from which we reconstructed the CC volume of size $40 \times 40 \times 40$ with the NN and tricubic filter. Figure 4 shows volume renderings of these spheres after 30 iterations and it is apparent that the reconstruction of the spheres using NN interpolation in the EM algorithm is more accurate. The spheres are more symmetric whereas the tricubic tomographic reconstruction is more irregular.

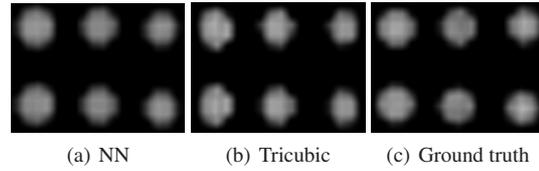


Figure 4: Small spheres reconstructed using (a) NN interpolation, and (b) the tricubic B-spline.

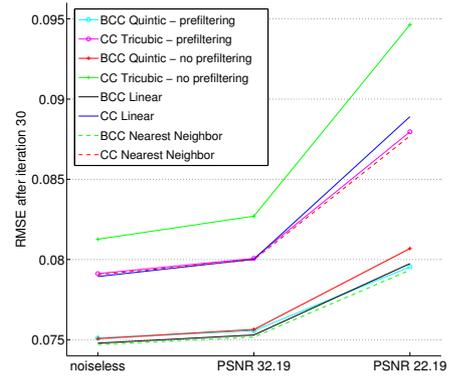


Figure 5: RMSE after iteration 30 for different noise levels and filters.

5.3. Different Lattices

Figure 5 already indicates the superiority of the BCC lattice since all four BCC reconstruction kernels show better error behavior than their CC counterparts. A possible explanation is the more isotropic topology of the BCC lattice.

The convergence of the EM algorithm (using NN filtering) for both lattices is illustrated in Figure 6 by the RMSE curves for all three noise levels between iterations 10 and 30. Iterations 1 to 9 were omitted for better readability. For each noise level, the BCC lattice outperforms the CC lattice (compare the dashed, solid, and “circle” curves).

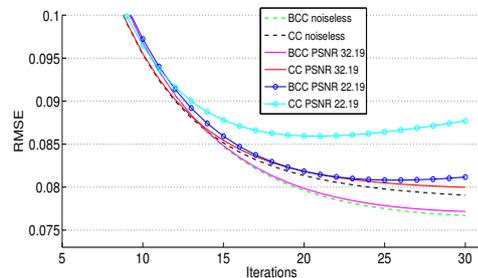


Figure 6: RMSE curves for iteration 10 to 30 for three different noise levels (noiseless, PSNR 32.19, PSNR 22.19) for the BCC and CC lattice.

The numerical differences demonstrated by the RMSE

curves are supported by visual results. In the following, we used the projections with a PSNR of 32.19 but similar results were obtained for noiseless projections and noisy projections with a PSNR of 22.19 and are presented in total in the file *sl_results.pdf* of the supplementary material.

Figure 8 shows two slices from the reconstructed volumes: (a) shows a 128×128 slice from the reconstructed CC volume after 30 iterations and (b) and (c) show the corresponding slice (100×100) from the reconstructed BCC volume. Note that although the BCC slice has only 100×100 pixels, the physical size of the slice is the same as the size of the CC slice. The reason for this is the different topology of the BCC lattice. To avoid confusion we therefore show the BCC slice with the same pixel size (b) and the same physical size (c) as the CC slice. In other words, (c) is just an upscaled version of (b). (d-f) show the corresponding slices of the ground truths (CC and BCC). When comparing the estimate of the CC volume (a) to the estimate of the BCC volume (b, c) the visual difference is noticeable: The BCC slice (b, c) shows less noise than its CC counterpart in (a). The last row (g, h) shows the intensity values indicated by the red lines in the middle row. The dashed black lines indicate the ground truth. Again, (h) shows less noise than (g) and demonstrates the better noise suppression of the BCC lattice. As a quantitative measure of noise suppression we computed the variance in the homogeneous dark grey region of the SL phantom. The lower the variance the better the noise suppression: For the BCC lattice the variance in the dark grey region is 0.000277 and for the CC lattice 0.000413, which shows that the noise suppression for the BCC lattice is almost twice as good as for the CC lattice.

The presented experiments were also performed with a BCC lattice of size $91 \times 91 \times 182$ which is 30% smaller than the CC lattice. A BCC lattice of this size can store the same information as the 128^3 CC lattice and the results in the supplementary file *sl_results.pdf* show that even this BCC lattice performs better than the CC lattice.

5.4. Real-World Data Experiments

We demonstrate the feasibility of tomographic reconstruction on BCC lattices using the GPU by reconstructing a volume from real-world data. We use projections of a mouse embryo that were acquired at the Max-Planck-Institute for Molecular Genetics using OPT. OPT is a method used to capture objects of the size of 1 to 10 mm diameter [SAP*02] at high spatial resolution and is employed for three-dimensional imaging of small biological specimen with optical light. Thus, different wavelengths (i.e., color) can be captured. Applications of OPT are mainly in the field of molecular biology and include gene-expression analysis, screening of abnormal anatomy or histology, or pinpointing cells within a tissue [Sha04].

We used 400 scalar OPT scans of a mouse embryo where each projection has a resolution of 471×696 pixels. Our

GPU implementation is able to reconstruct a high-resolution volume ($321 \times 474 \times 642$ BCC samples, see Figure 7) within less than one hour (30 iterations). Our reference CPU implementation requires seven days employing eight cores.



Figure 7: Volume rendering of the mouse embryo which was acquired from 400 OPT scans on a BCC lattice ($321 \times 474 \times 642$). Red areas indicate the most dense tissue.

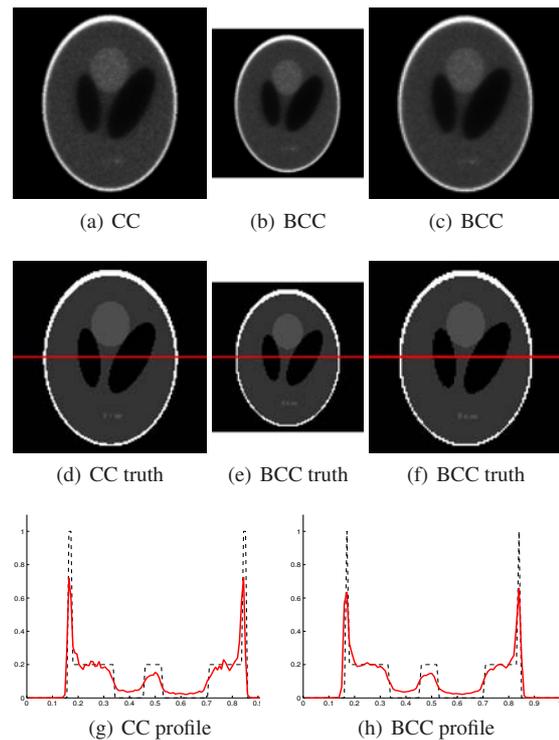


Figure 8: (a): CC slice of the reconstructed volume. (b): corresponding BCC slice. (c): upscaled version of (b). Second row: corresponding ground truths. Last row: intensity profiles indicated by the red lines in (d-f).

6. Conclusion and Future Work

We have established in a mathematical and empirical way the connection between the standard EM algorithm and the

continuous-domain volume rendering framework using NN interpolation. This contradicts the intuition that higher-order filters lead to higher accuracy; i.e., for the EM algorithm to work best with volume rendering techniques, one has to use NN interpolation. Higher-order reconstruction filters are incompatible. Note that NN interpolation is also the fastest reconstruction scheme due to its small and compact support.

Furthermore, we have demonstrated that volumetric reconstruction on the BCC lattice is more accurate and better suited for noise suppression than traditional reconstruction on CC lattices. The advantage of our method is that the acquired projections are still on 2D Cartesian lattices and therefore acquisition devices do not need to be changed. This result opens the possibility to a more wide-spread use of optimal sampling lattices in the areas of computed tomography and volume rendering.

So far, we have not modeled effects such as scatter correction, collimator blur, and attenuation correction. However, we assume that NN interpolation will also improve results because of its fundamental connection to the EM algorithm. The investigation of these effects are subject to future work.

Furthermore, we plan to investigate the possibilities of extending the EM algorithm for higher-order filters by matching the forward and back-projection for arbitrary reconstruction kernels. This could enable us to use the advantages of higher-order filters for CT.

References

- [BDCM05] BERGNER S., DAGENAIS E., CELLER A., MÖLLER T.: Using the physically-based rendering toolkit for medical reconstruction. *Proceedings of 2005 IEEE Nuclear Science Symposium and Medical Imaging Conference* (October 2005), 3022–3026.
- [CH06] CSÉBFALVI B., HADWIGER M.: Prefiltered B-spline reconstruction for hardware-accelerated rendering of optimally sampled volumetric data. *Vision, Modeling and Visualization* (2006), 325–332.
- [CM03] CHIDLOW K., MÖLLER T.: Rapid emission tomography reconstruction. *Workshop on Volume Graphics (VG03)* (July 2003), 15–26.
- [Csé05] CSÉBFALVI B.: Prefiltered Gaussian reconstruction for high-quality rendering of volumetric data sampled on a body-centered cubic grid. *IEEE Visualization* (2005), 40–48.
- [Csé08] CSÉBFALVI B.: BCC-splines: Generalization of B-splines for the body-centered cubic lattice. *Winter School of Computer Graphics* (2008).
- [Dea83] DEANS S. R.: The radon transform and some of its applications. *A Wiley-Interscience Publication, New York* (1983).
- [EDM04] ENTEZARI A., DYER R., MÖLLER T.: Linear and cubic box splines for the body centered cubic lattice. *Proceedings of IEEE Visualization* (2004), 11–18.
- [EVM08] ENTEZARI A., VILLE D. V. D., MÖLLER T.: Practical box splines for reconstruction on the body centered cubic lattice. *IEEE Transactions on Visualization and Computer Graphics* 14, 2 (2008), 313–328.
- [FDK84] FELDKAMP L. A., DAVIS L., KRESS J. W.: Practical cone beam algorithm. *J. Optical Society of America* 1, 6 (6 1984), 612–619.
- [FEMV09] FINKBEINER B., ENTEZARI A., MÖLLER T., VILLE D. V. D.: *Efficient Volume Rendering on the Body Centered Cubic Lattice Using Box Splines*. Tech. Rep. TR 2009-04, Simon Fraser University, 3 2009.
- [GBH70] GORDON R., BENDER R., HERMAN G. T.: Algebraic reconstruction techniques (ART) for three-dimensional electron microscopy and X-ray photography. *Journal of Theoretical Biology* 29, 3 (1970), 471–81.
- [KS88] KAK A. C., SLANEY M.: *Principles of Computerized Tomographic Imaging*. IEEE Press, 1988.
- [LC84] LANGE K., CARSON R.: EM reconstruction algorithms for emission and transmission tomography. *Journal of Computer Assisted Tomography* 8, 2 (1984), 306–16.
- [LM03] LEWITT R. M., MATEJ S.: Overview of methods for image reconstruction from projections in emission computed tomography. *Proceedings of the IEEE* 91, 10 (2003), 1588–1611.
- [Miz73] MIZOHATA S.: *The Theory of Partial Differential Equations*. Cambridge University Press, 1973.
- [ML95] MATEJ S., LEWITT R. M.: Efficient 3D grids for image reconstruction using spherically-symmetric volume elements. *IEEE Transactions on Nuclear Science* 42, 4 (1995), 1361–1370.
- [MY96] MUELLER K., YAGEL R.: The use of dodecahedral grids to improve the efficiency of the algebraic reconstruction technique (ART). *Annals Biomedical Engineering, Special issue, 1996 Annual Conference of the Biomedical Engineering Society* (1996), S–66.
- [SAP*02] SHARPE J., AHLGREN U., PERRY P., HILL B., ROSS A., HECKSHER-SØRENSEN J., BALDOCK R., DAVIDSON D.: Optical projection tomography as a tool for 3D microscopy and gene expression studies. *Science* 296 (2002), 541–545.
- [Sha04] SHARPE J.: Optical projection tomography. *Annual Review of Biomedical Engineering* 6 (August 2004), 209–228.
- [Sit07] SITEK A.: Programming and medical applications using graphics hardware. *Nuclear Science Symposium and Medical Imaging Conference* (2007).
- [SL74] SHEPP L. A., LOGAN B. F.: Reconstructing interior head tissue from X-ray transmissions. *IEEE Transactions on Nuclear Science* 21, 1 (1974), 228–236.
- [SV82] SHEPP L. A., VARDI Y.: Maximum likelihood reconstruction for emission tomography. *IEEE Transactions Medical Imaging* 1, 2 (1982), 113–122.
- [TMG01] THEUSSL T., MÖLLER T., GRÖLLER E.: Optimal regular volume sampling. *Proceedings of IEEE Visualization* (2001), 91–98.
- [XM07a] XU F., MUELLER K.: Applications of optimal sampling lattices for volume acquisition via 3D computed tomography. *Volume Graphics Symposium* (2007), 57–63.
- [XM07b] XU F., MUELLER K.: Real-time 3D computed tomographic reconstruction using commodity graphics hardware. *Physics in Medicine & Biology* 52 (2007), 3405–3419.
- [Xu07] XU F.: *Accelerating Computed Tomography on Graphics Hardware*. PhD thesis, Stony Brook University, Stony Brook, New York, 2007.
- [ZG00] ZENG G. L., GULLBERG G. T.: Unmatched projector/backprojector pairs in an iterative reconstruction algorithm. *IEEE Transactions on Medical Imaging* 19, 5 (2000), 548–555.