# A COMPARATIVE STUDY OF POPULAR INTERPOLATION AND INTEGRATION METHODS FOR USE IN COMPUTED TOMOGRAPHY

*Fang Xu and Klaus Mueller*

Center for Visual Computing     Computer Science     Stony Brook University

## ABSTRACT

We compare various popular methods available for projection and backprojection in CT. Assuming linear rays and a simple density integration along them, we consider both line- and area-based methods. Here, two key components govern the quality of a projection result, given the discrete nature of the data and reconstruction result: interpolation and integration. Both of these are studied here. In order to separate these fundamental issues from those related to perspective fan and cone-beam effects, we restrict ourselves to a parallel-beam projection geometry. We also compare these different methods in light of a possible efficient implementation on programmable commodity graphics hardware (GPUs). To this end, we propose a new method for interpolation based on hexagonal subsampling, which achieves superior results. In order to achieve a data-independent comparison, we employ a dataset of very high and uniform frequency content, the so-called Marschner-Lobb dataset.

## 1. INTRODUCTION

A plethora of methods for projection and backprojection methods has become available for CT through the past three decades. One way to distinguish these is in the manner they relate the data to be generated (the reconstruction) to the data provided (the scanner data). With 3D reconstruction in mind, we shall refer to the former as the *volume*, composed of *voxels*, and the latter as (projection) *image*, composed of *pixels*. While analytical CT algorithms, such the approximation by Feldkamp (FDK) [5] mainly involve a backprojection, iterative algorithms, such as SART [1] and EM [11], involve both projections and backprojections. In most cases, backprojection is simply the inverse of a projection, unless unmatched projector pairs are used [15]. The target of a projection is a (projection) image $P$, while the target of a backprojection is the volume $V$. Iterative algorithms work by projecting $V$ to create an image estimate $P'$ which is then compared with the acquired image $P$ and a correction image $P''=s(P-P')$ is backprojected -- $s$ being a scale factor -- which brings $V$ closer to an accurate estimate. The projection operator is written as: $P=WV$ and the backprojection operator is written as $V=W^{-1}P$. Here, $P$ is a $N^2$-long vector of pixels, $V$ is a $N^3$-long vector of voxels, and $W$ is a $N^2$ x $N^3$ weight matrix. In the general case, in particular for 3D cone-beam reconstruction, the size of $W$ prohibits its storage, and thus its elements have to be computed on the fly. In fact, $W$ is a sparse matrix, since generally a pixel in $P$ traces out a ray line or a ray beam, which only cover a subset of connected voxels (see Fig. 2).

The remaining issue is how the voxels that fall within the influence of this beam or line contribute to the pixel emitting it. This is where the various methods differ, and they do so in three, mostly orthogonal ways:

- Either they assume that the voxels are solid blocks or that the voxels are infinitesimal thin spikes (or sample points). In fact, this just leads to different kernels (a box for the former).
- Either they trace rays emerging from the pixels, or a few subpixels within a pixel, or they trace beams, usually bounded by the pixel (detector) boundaries.
- Either they trace the rays (or beams) across the volume from the pixels or they project the voxels onto the image plane.

Finally, they either perform a piecewise constant, piecewise linear, or continuous integration. We should note that not all of the combinations have actually been proposed and implemented, and we shall restrict our discussion to only those that have been.

We have studied these methods in a common framework designed to perform these tests in a strict signal processing context. This is justified by the fact that interpolation, and to a somewhat lesser extent, integration, are operations that seek to estimate a continuous function from a set of discrete sample points. While during data acquisition (the X-ray projection in the scanner) the rays have traversed an object that was naturally defined everywhere, this is not the case in the simulation of this process, where the object under reconstruction is only defined at discrete data points. The circumstance that a collection of rays that is averaged (before the log-operation) within a certain detector bin, poses another challenge for this simulation during reconstruction. A faithful simulation of the projection process, however, is key to a faithful reconstruction, and we shall compare the different methods with regards to this viewpoint (neglecting any other adverse effects, such as scattering, beam hardening, and polychromaiticity). But ultimately, we are interested in a faithful reconstruction, and this motivates an extension of this comparative study to a reconstruction task. In order to avoid possibly confounding perspective effects implied by fan or cone-beam, we perform our study in a parallel-beam reconstruction scenario, using our strict signal processing-motivated test function. However, we also conduct our test on a more realistic dataset bearing anatomical constraints, here, a CT head.

Our paper is structured as follows. Section 2 discusses previous work, while Section 3 provides a deeper discussion of the different interpolation and integration methods we have studied and an attempt to compare these in a common framework. Section 4 presents the test configuration and the results we have obtained. Section 5 presents some concluding arguments and also puts the study in perspective to the emerging trend of accelerating CT on commodity graphics hardware [14], which puts certain constraints on the interpolation schemes that can be efficiently used.

## 2. RELATED WORK

A wealth of publications exist that discuss and compare interpolation filters, and the limited space only permits to mention a few of these here. Thévenaz et al. [13] provide a comprehensive study on interpolation filters using frequency domain arguments, while

Möller et al. [10] view this task from a numerical standpoint, via a Taylor series expansion. Siddon [12], Joseph [7], Herman [6], and Lewitt [8] have described interpolation and integration mechanisms that are frequently used in CT today. Other, more recent papers, have enhanced and augmented these basic approaches, and the reader is referred to [4] for a more complete list of references. CT reconstruction is in some ways similar to volume rendering, where the goal is to project a volume dataset for visualization purposes. A seminal paper with respect to the study and comparison of interpolation filters in the context of volume rendering was presented by Marschner and Lobb [9]. To conduct their comparison they designed a rigorous test function, now known as the Marschner-Lobb (ML) function, which has a near-uniform frequency content that extends very close to the Nyquist rate and is contained within that interval to 99.8%. Its equation is given here:

$$\rho(x,y,z) = \frac{(1-\sin(\pi z/2)+\alpha(1+\rho_r(\sqrt{x^2+y^2})))}{2(1+\alpha)}$$
$$\rho_r(r) = \cos\left(2\pi f_M \cos\left(\frac{\pi r}{2}\right)\right) \quad (1)$$

Here, $\alpha$=0.25 and $f_M$ controls the frequency bandwidth for a given volume size. We set it $f_M$=18, which provides the desired full frequency range for a volume size of $128^3$ when $-1 < x, y, z < 1$.

A volume iso-surface rendering of the function, sampled into a volume of $40^3$ with $f_M$=6 and rendered at an iso-value of 0.5 is shown in Fig. 1. The interpolation filter used in the rendering (using raycasting) was of good quality and only caused little aliasing, as is evident from the modest deformations at the sinusoid rims. Since in CT we are reconstructing an *estimate* of the volume dataset from a set of near-analytical projections, we find it more useful to compare the reconstructed dataset with the true function in a numerical sense, via its RMS, and not via a visualization, which was done in the original ML work.

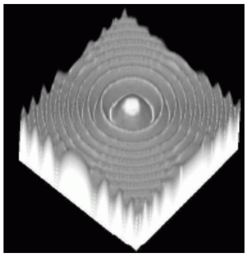


Figure 1: The Marschner-Lobb function

## 3. METHODS

Fig. 2 illustrates various interpolation and integration strategies. Although we only show the interpolation for the 2D case, for 3D rendering, the drawings would extend into 3D (which turns every linear interpolation into a bilinear interpolation).

- The *slice-interpolated* method uses bilinear interpolation within each slice and integrates the results in a trapezoidal fashion. This is a 3D extension to Joseph's method [7] and is also the method our GPU projective-texture renderer employs [14]. Depending on the viewing angle, the sampling rate along the ray varies ($1...\sqrt{2}$), and so does the integration interval. One way to increase the integration quality is to use a more sophisticated quadrature method, such as Simpson's method along with an intermediate sample.
- The *grid-interpolated* scheme can space the samples at the same distance, independent of viewing angle, usually at 1.0. Interpolation is commonly performed with a trilinear filter, and integration is again according to the trapezoidal rule.
- The *box-line-integrated* scheme was proposed by Siddon [12]. This corresponds to a box interpolation filter (kernel) and a continuous integration (assuming a piecewise constant signal).
- The *RBF-line-integrated* method (RBF=Radial Basis Function) was proposed by Lewitt [8] and is used in conjunction with a forward projection (splatting) of a pre-integrated kernel function. The integration is continuous, similar Siddon's method, since the kernel (usually a Gaussian or a Bessel function) is superior to a box, but also much wider, taking longer to project.
- The *box-beam-integrated* method is an extension to Siddon's method, where now the entire box volume that falls inside the extended pixel boundaries is added to the integral. It still assumes an inferior box kernel, but it captures the width-integrating nature of the X-ray beam better than a line. The distance-driven method of [4] works along these lines.
- Not shown is the so-called *pixel-driven* method (we call it *voxel-driven*) [6], where each voxel is projected to the screen and its contribution written to the nearest pixel or distributed with bilinear weighting among the four pixels in the screen square. The latter is similar to the splatting of a bilinear kernel.

Since the acquired data are proportional to the average to the beam of rays that end in a given detector bin, a beam-integrated method is at least potentially a better choice. We have just discussed the box-beam-integrated (distance-driven) method, which uses an underlying box filter. The task of computing the solid encapsulated within a beam is feasible to compute. It is more difficult to do so

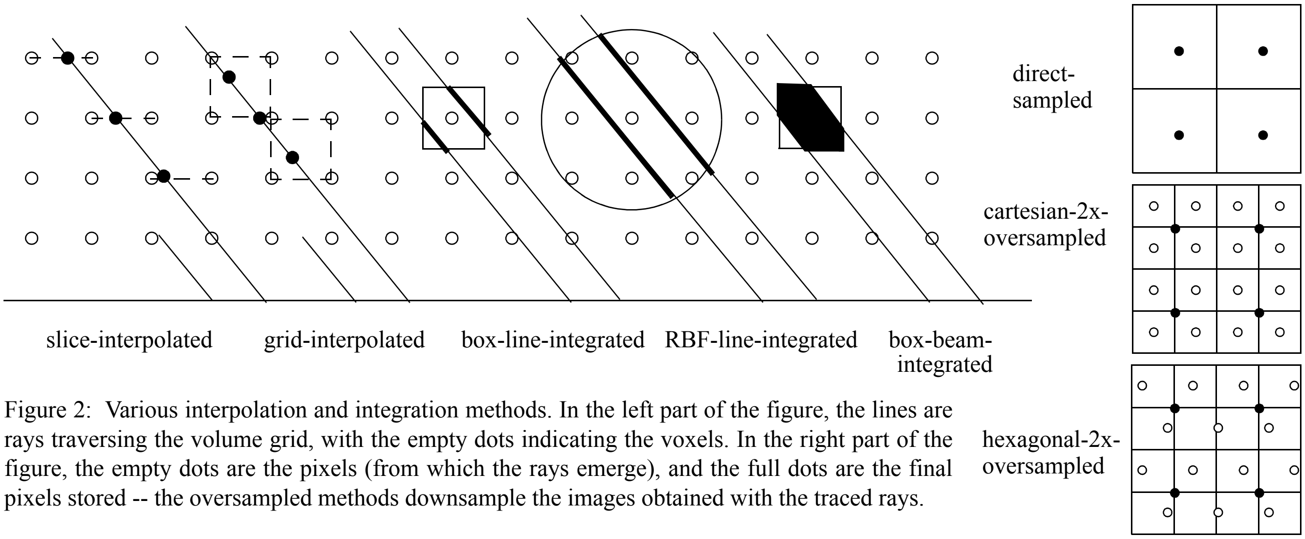


Figure 2: Various interpolation and integration methods. In the left part of the figure, the lines are rays traversing the volume grid, with the empty dots indicating the voxels. In the right part of the figure, the empty dots are the pixels (from which the rays emerge), and the full dots are the final pixels stored -- the oversampled methods downsample the images obtained with the traced rays.

for higher-order functions, although it can be done with the radially-symmetric functions, such as the RBF method, using a Summed-Area-Table [3]. An approximation of the beam-tracing that can use better interpolation kernels is to trace extra rays and then downsample the result, using some sort of lowpass filter tuned to the sampling rate of the output grid. The most obvious way is the *cartesian-2x-oversampled* scheme shown on the right side of Fig. 2. However, this requires the tracing of 4-times the number of original rays, which can be slow. A better solution in this regard is what we call the *hexagonal-2x-oversampled* scheme, which we propose as a new contribution in this paper. According to lattice theory [2], it can hold the same frequency content than the equivalent cartesian grid. For example, assuming a unit grid spacing for the *direct-sampled* case, the hexagonal grid would sample at $1/\sqrt{3}$ along $x$ and at 0.5 along $y$. This would require 29.3% less samples, that is, for a $100^2$ grid, the cartesian scheme would require 40,000 rays, compared to 28,280 rays for the hexagonal scheme.

## 4. RESULTS

We conducted both a projection and a reconstruction study to evaluate the performance of various interpolation and integration methods. All results are produced on a 2.2GHz dual core AMD Athlon PC with 1GB RAM.

### 4.1. Projection Study

We generated both line-integrated and beam-integrated reference images from the ML dataset. The latter are produced by calculating 16 sub-rays for each detector element and box-filtering the values. Both types of reference images are computed by sampling the object function with a dense step size of 0.2 unit distances. A discretized 3D floating point dataset ($128^3$ resolution) sampled from the object function is used to generate all other projections. We measure the RMS error from different viewing angles, particularly focusing into oblique views in the range of 40° to 45°, where slice-integrated schemes are prone to have artifacts due to their insufficient sampling rate along the ray direction from those views. Note that we compare all strategies in the beam-integrated study, but only line-integrated schemes in the other.

Fig. 3(a) and (b) plot the evaluation results from each strategy against line-integrated and beam-integrated reference images, respectively. We observed that in the line-integrated scanner projections study, Siddon's method (box-line-integrated scheme) produces overall better projections than all other methods, which we believe is partly due to its more accurate continuous integration path. Slice-interpolated approaches, while having less sampling density along the ray direction compared to grid-interpolated methods, have nevertheless similar image quality. The voxel-driven scheme with bilinear kernel has similar performance but suffers a severe deterioration in image quality at angles close to 45°. The blurry effect from the RBF-line-integrated method leads to a low score when compared to line-integrated scanner projections. Finally, we also observe that Simpson's integration does not help improve image quality for slice-integrated schemes.

While in the beam-integrated study, the RBF-line-integrated method using the Bessel kernel achieves the best image quality, slice-integrated methods and grid-integrated approach perform as just as good as Siddon's method. Also it helps to generate more accurate projections by computing extra rays for the ray-driven methods, where hexagonal-2x-oversampled and trilinear-2x-oversampled schemes surpass the cartesian-2x-oversampled scheme. If the trapezoidal integration is replaced by Simpson's rule in slice-integrated and hexagonal-2x-oversampled schemes, further improvement on image quality can be obtained. However, oversampling in the object space via computing intermediate slices has little impact for slice-interpolated approaches. Similarly, the voxel-driven scheme with the bilinear kernel yields results close in quality to Siddon's method, except at views around 45°.

### 4.2. Reconstruction Study

We employed the Simultaneous Algebraic Reconstruction Technique (SART) as our test algorithm to perform this study. In addition to the ML dataset, a human CT skull volume is also used for evaluation (the chart for skull volume is not shown here due to space constraint). Both datasets are discretized into a grid of $128^3$ , while 80 views of reference images of $128^2$ detector elements are acquired uniformly within 360 degrees. A relaxation parameter of 0.1 and 20 iterations are applied throughout the experiment. Both line-integrated and beam-integrated reference images are generated for the ML study, while only line-integrated references are derived for the human CT skull study. Fig 3. (c) and (d) demonstrate the performance of all methods for the ML dataset in the line- and beam-integrated settings, respectively.

We notice that the slice-integrated schemes have at least comparable performance with Siddon's method and the grid-integrated method, in terms of both reconstruction quality and convergence speed. Similar to the projection study, given beam-integrated scanner projections, over-sampling on the detector plane helps to reduce the error, and here the hexagonal-2x-oversampled scheme outperforms the cartesian-2x-oversampled scheme, and even the RBF-line-integrated method. In contrast, there does not appear to be an advantage in applying Simpson's rule over the trapezoidal integration.

## 5. CONCLUSIONS

Our evaluation study demonstrates that the slice-interpolated schemes have projection and reconstruction performances comparable to those of the grid-interpolated and box-line-integrated schemes, despite the reduced and non-uniform sampling rates. The hexagonal-2x-oversampled scheme we have proposed in this paper, in fact performs better than the cartesian-2x-oversampled scheme, yet costs less computational effort.

As mentioned above, one goal of our study was to also provide some guidance with respect to the quality of interpolation and integration schemes commonly used in fast GPU-based CT implementations. Here, the slice-interpolated schemes are the most convenient to use in this setting, and we have just seen that they do not lag behind in quality. Thus, this study provides ample assurance for this choice.

## 6. ACKNOWLEDGEMENTS

This work was partially funded by NIH grant R21 EB004099-01 and Agard Lab, Department of Biochemistry and Biophysics, University of California at San Francisco.

## REFERENCES

[1] A. Andersen A. Kak, “Simultaneous Algebraic Reconstruction Technique (SART): a superior implementation of the ART algorithm,” *Ultrason. Imag.*, 6:81–94, 1984.

[2] J. Conway and N. Sloane, *Sphere Packings, Lattices and Groups*, 2nd edition, Springer Verlag, 1993.

[3] F. Crow, “Summed-area tables for texture mapping,” *Proc. Siggraph 84*, pp. 207-212, 1984.

[4] B. De Man, S. Basu, “Distance-driven projection and backprojection in three dimension,” *Phys. Med. & Biol.,* 49:2463–2475, 2004.

[5] L. Feldkamp, L. Davis, J. Kress, “Practical cone beam algorithm,” *J. Opt. Soc. America*, pp. 612–619, 1984.

[6] G. Herman, *Image Reconstruction from Projections*. Academic Press, 1980.

[7] P. Joseph, “An improved algorithm for reprojecting rays through pixel images,” *IEEE Trans. Med. Imag.* 1 192–6, 1983.

[8] R. Lewitt, “Alternatives to voxels for image representation in iterative reconstruction algorithms,” *Phys. Med. & Biol.,* 37:705–716, 1992.

[9] S. Marschner, R. Lobb, "An Evaluation of Reconstruction Filters for Volume Rendering," *IEEE Visualization 1994*, pp. 100-107, 1994.

[10] T. Möller, R. Machiraju, K. Mueller, Roni Yagel, "Evaluation and Design of Filters Using a Taylor Series Expansion," *IEEE Trans. Visual. Comp. Graph.*, 3(2):184-199, 1997.

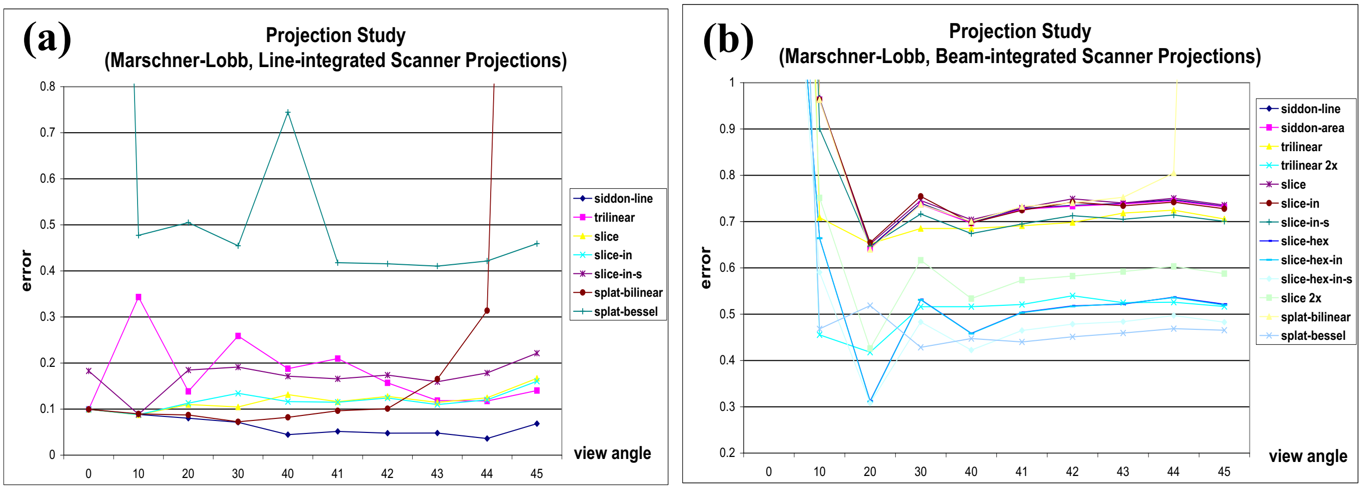
[11] L. Shepp and Y. Vardi, “Maximum likelihood reconstruction for emission tomography,” *IEEE Trans. Med. Imag.*, 1:113–122, 1982.

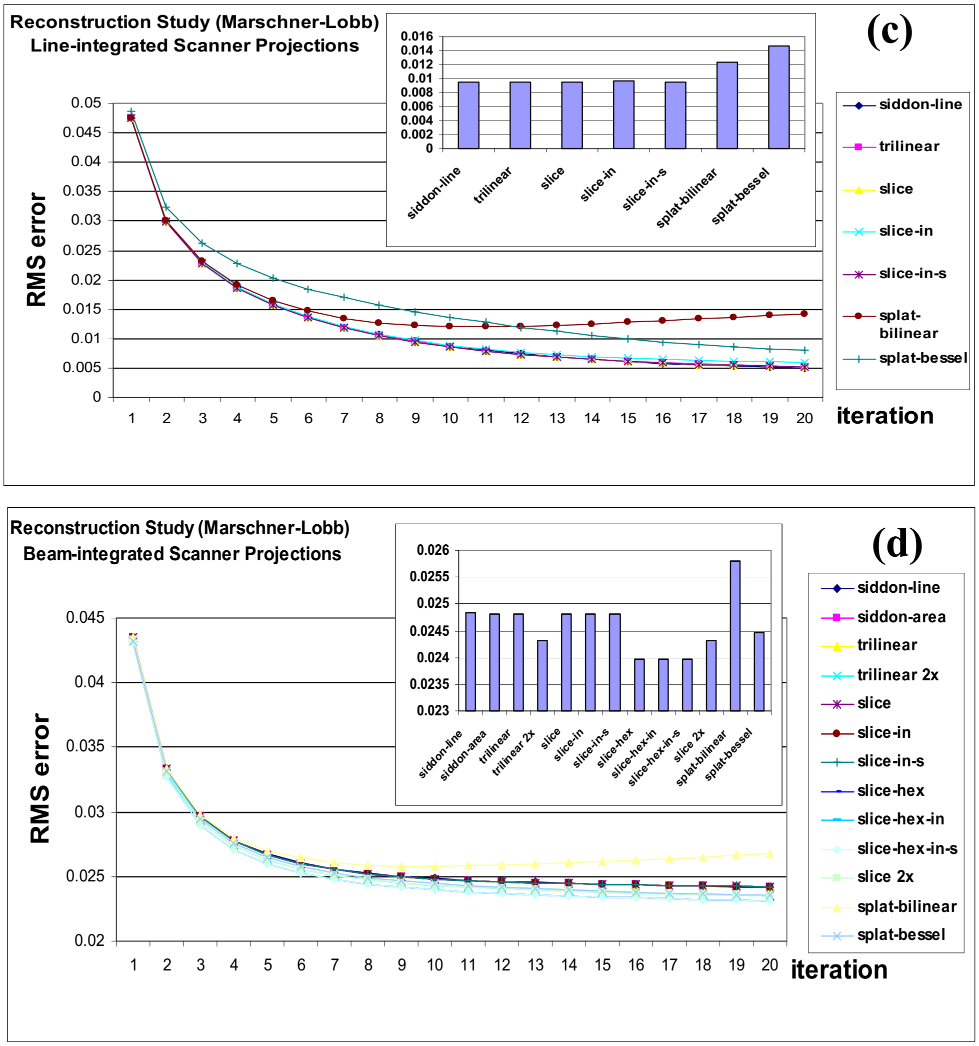
[12] R. Siddon. “Fast calculation of the exact radiological path length for a three-dimensional CT array,” *Med. Phys.* 12**:**252–255, 1985.

[13] P. Thévenaz, T. Blu, M. Unser, "Interpolation revisited," *IEEE Trans. Med. Imag.*, 19(7):739-758, 2000

[14] F. Xu, K. Mueller, “Accelerating popular tomographic reconstruction algorithms on commodity PC graphics hardware,” *IEEE Trans. Nucl. Sci.*, 52(3):654-663, 2005.

[15] G. Zeng, G. Gullberg, "Unmatched projector/backprojector pairs in an iterative reconstruction algorithm," *IEEE Trans. Med. Imag.* 19(5): 548-555, 2000





Note: we use the term *scanner projections (SP)* for the data that would be commonly acquired by means of a scanner (the raw data). For the purpose of this study, we simulate this process. The same data are also used in the Projection Study, for the comparisons.

Figure 3: Evaluation Study: (a) projection study with line-integrated scanner projections (SP). (b) projection study with beam-integrated SP. (c) reconstruction study with line-integrated SP. (d) reconstruction study with beam-integrated SP.

**siddon-line**: box-line-integrated
**siddon-area**: box-beam-integrated
**trilinear**: grid-interpolated
**slice**: slice-interpolated
**splat**: voxel-driven
**-hex**: hexagonal grid
**-2x**: double number of rays
**-bilinear**: bilinear kernel
**-bessel**: Bessel kernel
**-in**: intermediate slices
**-s**: Simpson’s integration

The inset graphs in (c) and (d) plot the reconstruction RMS error after 10 iteration for each scheme.