Numerical Evaluation on the Derivative Operator of Filtered Backprojection Image Reconstruction

Ibrahim Mkusa[1], Yanbo Zhang[1], Morteza Salehjahromi[1] and Hengyong Yu[1,*]
1. Department of Electrical and Computer Engineering, University of Massachusetts Lowell, Lowell, U.S.A
*. Corresponding author, Email: Hengyong-yu@ieee.org

***Abstract*— The analytic filtered backprojection algorithms are widely used in the computed tomography field. It is well known that the filtering step is implemented either by a ramp kernel function or by combining the derivative and Hilbert filtering operations. Although the derivative is a local operator, the corresponding discrete kernel function has infinite support. It is practical to compute derivative using some numerical methods, such as forward, backward, central differences and spline fitting. In this paper, we numerically evaluate the performance of different numerical methods for computing derivative by fixing other factors. By quantitatively analyzing the reconstructed image quality, we look into tradeoffs between image quality indexes when each type of derivative is used. Results show that, the central difference approximation derivatives produce the most accurate images with lower noise and spatial resolution, whereas the spline fitting leads to the highest spatial resolution with higher noise.**

Keywords—Filtered Backprojection; Derivation Operator; Numerical Simulation; Central Difference; Spline Fitting

I. INTRODUCTION

In the computed tomography (CT) field, the analytic Filtered Backprojection (FBP) algorithms are widely adopted for reconstructing images. They are preferred due to their fast and efficient implementations over the competitive iterative algorithms like simultaneous algebraic reconstruction technique (SART), which assume an image as an array of unknowns and solve it by setting up equations in terms of projections. The accuracy and speed of iterative algorithms come at a computational cost. Projections are essentially modelled as a collection of line integrals through an object (the image) at certain angles [1]. For a simulated object like the popular Shepp-Logan head phantom [2], its projections can be analytically computed by taking the integrals along lines oriented at a certain angle. For real objects, like the human brain, we can't measure line integrals directly. However, we can indirectly measure them by taking advantage of the fact that x-rays are attenuated when they pass through matter. The photoelectric and Compton effects cause x-rays to decrease in energy, and in number of photons. Comparing energy and photon count in the incident and emergent ray allows us to determine the line integrals by performing logarithmic operation based on the so-called Beer's law. The integrals are of the x-ray attenuation coefficient along points through the line. It should be noted that this only works for monochromatic x-rays. Polychromatic x-rays, the most common form of x-rays in clinical practice, consist of a large spectrum of energies and they require more involved calculations to approximate [1]. The logarithmic value of the ratio between the intensities of the original and received x-rays is considered as the projection. The reduction (or the attenuation) of the intensity of an x-ray as it passes through the object is what makes computed tomography possible for real objects. Although this model is a gross simplification, it will do for the purpose of this paper.

With projections in hand, CT reconstruction is essentially an inverse problem to obtain the original distribution of the attenuation coefficients (e.g. image). An FBP algorithm takes these projections as input and reconstructs images as output. For every projection view, its measurements are first weighted with appropriate function to account for redundancy, filtered, and finally copied through the entire image plane (backprojection). Redundancy happens because some line integrals are measured more than once. The filtering step in an FBP algorithm can be implemented in two main ways. One is to use the ramp filtering, and the other is to use the derivative followed by the Hilbert filtering [1, 3]. In this paper, we will focus on the later. In fact, more ways can be obtained by changing the order of the aforementioned steps. Each FBP algorithm can have more variations with different tradeoffs. Depending on the x-ray beam shape, we have parallel beam or fan-beam geometry. Fan-beam employs a single x-ray source that spreads the x-rays in the shape of a fan, while parallel beam uses a single source that has to linearly scan over a length of a projection sending parallel rays one at a time. For the fan-beam geometry, depending on how the arrays of detectors are arranged, we have equiangular or equidistant detectors. Equidistant detector geometry means the sensors are positioned in equal distances from each other, while the equiangular detector geometry means the sensors are positioned at equal angles from each other. In practice, fan-beam is preferred for 2D CT reconstruction as it takes less time to obtain the projections [1,3]. Without loss

of generality, we will evaluate an FBP algorithm assuming fan-beam geometry and the equiangular detectors.

As the aforementioned, the filtering step is an important part of an FBP algorithm. It has been well established that a ramp filter kernel function can be used [2] and several numerical methods have been developed, such as R-L filter [13], S-L filter [14], *etc*. From the point view of Fourier analysis for signal processing, the ramp filter can be decomposed into the derivative operation and the Hilbert filter [3]. While there are lots of studies that have gone into the Hilbert filter and its discrete counterparts [4-5], as far as the authors know, there are no papers dedicated to the derivative operation in an FBP algorithm. This motivates us to perform a comprehensive evaluation on the derivative operation for the FBP algorithms. Because the discrete version of an ideal derivation operator has infinite support, our evaluation focuses on the derivative approximation. This includes the forward, backward, and central differences, spline interpolation, *etc*. We numerically evaluate these methods and assess their impacts on image quality of the reconstructed images. Our goal is to optimize those numerical methods and provide a practical guidance for derivation operation in an FBP algorithm. This can help to numerically optimize the engineering implementations for all FBP-type algorithms for better performance, particularly for quantitative analysis based applications (e.g. lung functional perfusion study). Because the derivative operation is also an important component in all backprojection filtration (BPF)-type algorithms, the results obtained should be directly applicable to all BPF-type algorithms, too.

The rest of the paper is organized as follow. In section 2, we describe the fan-beam equiangular imaging geometry, projection model and the corresponding FBP reconstruction formula. In section 3, we perform theoretical analysis on the discrete derivative operation and the approximation methods. In section 4, we report our experiment design and numerical results from simulated projected data of the 2D Shepp-Logan head phantom, as well as clinical data obtained from clinical CT scanners. In last section, we discuss some related issues and conclude this paper.

II. IMAGING GEOMETRY AND FILTERED BACKPROJECTION

To describe the FBP formula, we first need to describe the imaging geometry in which the projections are measured. The imaging geometry in this paper is depicted in Fig. 1 [4]. The figure shows an object centered at the origin of the Cartesian coordinate plane. An x-ray source assuming a point emits x-rays that illuminate the object in a fan like fashion. The distance from the ray source to the center of the object is called the focal length, denoted by D. The photons along the x-ray paths (projection rays) are received by the arced detector cells [4]. In practice, a detector cell integrates the energies of all the received photons. By comparing the energy integral to the incident ray, it gives us a measure of attenuation which is modeled by line integrals in theory [3]. At each point the detector cell receives all the photons along the corresponding x-ray path, the value of the line integral of the x-ray path is recorded. A collection of these values over the entire length of the detector make a projection taken at a view angle β. The projections are represented by $g(\gamma,\beta)$ which is a function of both β and γ, and γ is used for x-ray path. To locate each x-ray path, the angle γ is calculated between the projection x-ray path and the central x-ray path which runs from the focal spot to the origin. For the equiangular case, the angle $\Delta\gamma$ between two consecutive x-ray paths is known. If we know, for example, that our x-ray path is the third from the detector center, it's angle is simply $3 * \Delta\gamma$.

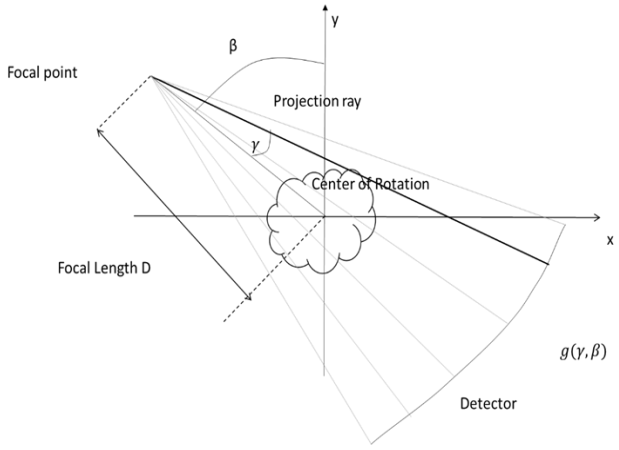


Fig. 1. Fan beam imaging geometry

As the aforementioned, FBP requires multiple projections taken at different β distributed around the object. After projection $g(\gamma,\beta_1)$ is taken, the x-ray source is rotated $\Delta\beta$ about the center of rotation in Fig. 1 to β_2 where projection $g(\gamma,\beta_2)$ is measured. The projections may be distributed $360°$ around the object, or less than it. A $360°$ scan of the object is called a full scan. Without loss of generality, the FBP algorithm chosen assumes a full scan in this work. Before presenting the full-scan fan-beam equiangular FBP formula, a word on the Fourier central slice theorem is pertinent. The Fourier slice theorem relates a projection to a parallel slice through the two dimensional (2D) Fourier transform of an image. It's the basis for the derivation of parallel beam FBP algorithms [3-4]. Unfortunately, due to the geometry of fan-beam and how the projections are taken, we cannot apply it directly to obtain fan-beam algorithms. We can, however, relate parallel beam algorithms to fan-beam algorithms by taking advantage of the fact the fan-beam projections can be rebinned into parallel beam geometry. We can either rebin all fan-beam x-rays to parallel beam and use parallel beam algorithms directly, or mathematically derive them from parallel ones. The former method is not used, because it introduces interpolation errors from the conversion. However, there are some studies using Laplacian kernel, also a local operator, plus inverse ramp kernel for truncation artifact correction.

The latter is in common use, and is the one used to derive this paper's formula [3-4].

The full scan equiangular FBP formula studied in this paper is expressed as

$$f(\boldsymbol{x}) = \frac{1}{2\pi}\int_0^{2\pi} \frac{1}{2\widetilde{D}(\beta,\boldsymbol{x})}\left(\int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} \frac{\partial g(\gamma,\beta)}{\partial\gamma}\frac{1}{\pi\sin(\tilde{\gamma}-\gamma)}d\gamma\right)d\beta \qquad (1)$$

where $f(\boldsymbol{x})$ is the reconstructed image at the position $\boldsymbol{x} = (x, y)$ in Cartesian coordinates [3-4], $g(\gamma,\beta)$ represents the projections, $\tilde{\gamma}$ is parameter for the ray path that passes through the point $\boldsymbol{x}$ for the view angle β, and $\widetilde{D}(\beta,\boldsymbol{x})$ is the distance between the focal spot of the view angle β and point $\boldsymbol{x}$ to be reconstructed. The partial derivative of the projections is first calculated with respect to γ. Then, it is filtered with the Hilbert filter $\frac{1}{\pi\sin(\tilde{\gamma}-\gamma)}$. On closer look, the Hilbert filtering is done by convolution with the partial derived projections. Noticing that the inner integral (i.e. convolution) assumes the equiangular detector span π radians, the integral goes from $\frac{-\pi}{2}$ to $\frac{\pi}{2}$. The Hilbert filter employs a sine operation which is a byproduct of the derivation of this fan-beam FBP from its parallel beam counterpart. The aforementioned fan-beam FBP algorithms are special cases of the parallel beam ones. The conversion involves changing variables, and the inclusion of Jacobian factor to the parallel beam formulas [3]. Every projection has to be weighted to the reconstructed image. This is necessary because of the fan-beam rebinning to parallel-beam. The weighting, $\frac{1}{2\widetilde{D}(\beta,\boldsymbol{x})}$, is a function of the projection view angle, and the point $\boldsymbol{x}$ in the image plane. $\widetilde{D}$ is the distance from the reconstruction point $\boldsymbol{x}$ to the x-ray source focal spot at angle β. After weighting, $f(\boldsymbol{x})$ represents our reconstructed image in Cartesian coordinates.

III. THEORETICAL ANALYSIS AND APPROXIMATION METHODS

Given a continuous function $f(x)$ where $x \in \mathbb{R}$, its derivative is $\frac{df(x)}{dx}$. $\frac{df(x)}{dx}$ at a value $x = x_o$ is defined as

$$\frac{df(x_o)}{dx} = \lim_{h\to 0}\frac{f(x_o+h)-f(x_o)}{h} \quad . \qquad (2)$$

The limit represents the slope of the tangent line at point x_o in the function $f(x)$. In practice, only discrete samples are available, and it is impossible to deal them as a continuous function. Let $f(x_n)$ represent the discrete function, where $x_n = n \times T$, $n \in \mathbb{Z}^+$ and T is the sampling period, sampled from the continuous function $f(x)$. $f(x_n)$ can be viewed as $f(x)$ multiplying by a periodic impulse train, $p(x)$ [15]. This is illustrated as

$$f(x_n) = f(x)p(x) \qquad (3)$$

where $p(x)$ is defined as

$$p(x) = \sum_{m=-\infty}^{\infty}\delta(x_n - mT) \qquad (4)$$

Here $\delta(x)$ is the unit impulse function. Equation (3) can be rewritten to account for the fact that multiplication of a function by a unit impulse samples the value of the function at the impulse's location [15].

$$f(x_n) = \sum_{m=-\infty}^{+\infty} f(mT)\delta(x_n - mT) \qquad (5)$$

If $f(x)$ is bandlimited to a frequency $\pm\omega_f$ and the sampling frequency $\omega_s = \frac{2\pi}{T}$ is such that $\omega_s > 2\omega_f$ (Nyquist Theorem), then $f(x_n)$ uniquely identifies $f(x)$ and they can be used for signal recovery [15]. There are several ways to do this among which band limited interpolation is theoretically exact. Band limited interpolation involves convolution with a sinc function $sinc(x)$.

$$f(x) = f(x_n) * sinc(x) \qquad (6)$$

where the "*" symbol refers to convolution and it not the usual multiplication. In digital signal processing, the sinc function is normalized such that $\int_{x=-\infty}^{\infty} sinc(x)dx = 1$, and is formally defined as

$$sinc(x) = \frac{\sin(\pi x)}{\pi x} \qquad (7)$$

Combining equations (5) and (6), it now gives us the interpolation formula of $f(x)$ such that

$$f(x) = \sum_{m=-\infty}^{\infty} f(mT)sinc(x - mT) \qquad (8)$$

Once we have described a basis for the discrete function, we can now proceed with analysis of the discrete operation. Applying continuous derivative to equation (8), it yields

$$\frac{df(x)}{dx} = \sum_{m=-\infty}^{\infty}\frac{f(mT)dsinc(x-mT)}{dx} \qquad (9)$$

Although the derivative is a local operation, equation (9) shows that the derivative at a position involves all the sample values. That is, the operation is nonlocal. This phenomenon can also be interpreted from equation (2). In practice, the exact solution to the limit is hard to calculate exactly by using equation (2) because it becomes a divide-by-zero problem when $h \to 0$. As a result, the derivative cannot be computed exactly, and has to be approximated this way. It must be pointed out that if the analytical form of the signal is known e.g. a quadratic $g(z) = z^2$, and the function is continuous and smooth, then the derivative can be computed exactly. However, there is no analytical form for most of the practical applications, such as CT reconstruction.

There are various approximation methods available. In this paper, we will evaluate four popular methods including finite differences and spline fitting. The finite difference methods are defined based on equation (2). For a point x_0 [6-8], we have three variants of finite differences (forward, backward and central)

$$\frac{df(x_0)}{dx} \approx \frac{f(x_0+T)-f(x_0)}{T} \qquad (10)$$

$$\frac{df(x_0)}{dx} \approx \frac{f(x_0)-f(x_0-T)}{T} \qquad (11)$$

$$\frac{df(x_0)}{dx} \approx \frac{f(x_0+T)-f(x_0-T)}{2T} \qquad (12)$$

The forward difference derivative approximation (FDDA), defined by equation (10), subtracts the value of the function at point x_0 from the function's value at a point a period ahead. This difference is then divided by the sampling period T to obtain the derivative. The backward difference derivative approximation (BDDA), defined by equation (11), is same as FDDA with one exception being in the numerator where the difference is obtained by subtracting the function's value at a period behind from its value at

point x_0. A thing to note is that derivatives from FDDA and BDDA are not at the exact points of interest but are shifted slightly to the right or left, respectively. Appropriate compensation has to be added in an application. To compensate for the shift seen in FDDA, 0.5 is subtracted from detector cell coordinate used to index into the projection. Similarly, 0.5 is added in the case of BDDA. This should be intuitive as FDDA or BDDA take points to the right or left, and at point of interest and averages them. The last of finite difference based methods is the central difference derivative approximation (CDDA) shown in (12). It takes the function's value at points a period ahead and below from the point of interest and divides it by double sampling period. It doubles the sampling period as its numerator spans twice the distance than FDDA and BDDA. It can be thought off as taking the average of the FDDA and BDDA at a point x_0. For this reason, CDDA gives a more accurate approximation of the derivative as it considers points around the point of interest [6-8].

The spline derivative approximation, as its name implies, uses a spline which is a piecewise smooth polynomial function. There are different types of splines, and the one employed in this paper is the third order natural cubic spline. This spline consists of piecewise cubic polynomials connecting a set of points. To make it smooth, the second derivative of each piecewise polynomial is set to be equal to every other at the endpoints. Given a one-dimensional set of I points $(y_0, y_1, \dots, y_I)$, the i^{th} piecewise polynomial connecting two points is given by [14]

$$Y_i(t) = a_i + b_i t + c_i t^2 + d_i t^3 \quad (13)$$

where $t \in [0,1]$ and $i = 0,1, \dots, N-1$. The remaining variables are defined in the following equations

$$Y_i(0) = y_i = a_i \quad (14)$$

$$Y_i(1) = y_{i+1} = a_i + b_i + c_i + d_i \quad (15)$$

Taking derivatives of $Y_i(t)$ to generate enough equations to solve yields

$$Y'_i(0) = b_i \quad (16)$$

$$Y'_i(1) = b_i + 2c_i + 3d_i \quad (17)$$

Equations (14) through (17) are enough to solve for the variables a_i, b_i, c_i, and d_i. As the aforementioned, the spline has to be smooth. Therefore, the following additional constraints are applied $Y_{i-1}(1) = y_i$, $Y'_{i-1}(1) = Y'_i(0)$, $Y_i(0) = y_i$, and $Y''_{i-1}(1) = Y''_i(0)$ for interior points in each interval.

Once the spline is calculated, the derivative at a point y is then the derivative of whatever piecewise polynomial at that point. The cubic spline derivative approximation (CSDA) method is focused on smoothness of the fitted spline rather than the accuracy of the derivative. As a result, a penalty to the derivative is anticipated.

IV. EXPERIMENTAL DESIGN AND RESULTS

A. Experimental Design

The FBP algorithm Equation (1) was implemented and ran using the MATLAB programming language on an Intel® Core™ i5-2540M CPU, 2.60 GHz and 6 GB RAM PC platform. The implementation was divided into three main parts: derivative of the projections, performing Hilbert filtering, and backprojection. The FDDA, BDDA, CDDA were straightforward to compute as they used simple arithmetic, whereas the SCDA required a spline fit. The MATLAB function "spline" was used to interpolate the splines between the points. The derivatives of the splines were calculated using "fnder" function, and evaluated at our points of interest using the "ppval" function [6-8]. To perform Hilbert filtering, the Hilbert kernel was first generated with respected to γ as shown in equation (1). The sine operation is then applied to γ to complete the filter. An important factor in the generation of the filtering kernel is its size with respect to the projections. Given projections are N elements big at a certain angle β, the discrete Hilbert filter kernel function must be $2N-1$. Performing discrete convolution of the discrete Hilbert filter with the projections yields a result thrice the size. In discrete convolution between two signals, the size of the resulting signal is the sum of the sizes of signals subtracted one [5]. The convolution was performed by calling the optimized convolution function in Matlab for fast computing.

The final step of the implementation is the backprojection. The final 2D image will be an M by N array. The reconstructed image will be centered at the origin of the coordinate plane similar to the object in Fig. 1. For every derived and filtered $g(\gamma, \beta)$ at angle β, we loop through the entire image array (initialized to zeros), and for every grid point its coordinate (x, y) in the projection view plane is computed. From them, the angle γ the point subtends is obtained. This angle is then converted to the detector cell coordinate. For FDDA and BDDA, pixel shift occurs in the reconstructed image as the derivative is offset by 0.5 and -0.5. To address this, 0.5 was subtracted or added from the detector cell coordinate, respectively. No adjustments were needed for the remaining two other methods. We then index into the derived and filtered $g(\gamma, \beta)$ with help of some linear interpolation, since the detector cell coordinate might not be exactly on a sampling point of γ. This value is added to the point (x, y) with the weighting $\frac{1}{2\tilde{D}}$. Once we've looped through every projection view β and do the same, the value of a point (x, y) is then the sum of every $g(\gamma, \beta)$ related to it. The entire reconstructed image is lastly divided by 2π to account for scale.

Experiments conducted were categorized into numerical simulations and preclinical applications. For numerical simulations, we ran the FBP on projections obtained from a 2D high-contrast Shepp-Logan phantom [2]. The Shepp-Logan phantom, a famous model of the human head, is used to obtain projections for testing the accuracy of image reconstruction algorithms. It is composed of 8 ellipses of varying centers, major and minor axis, orientation, and gray level. The ellipses model different tissues in the human head. Table 1 lists the parameters of the Shepp-Logan phantom. Columns a and b represent the major and minor

axis of the ellipses. Columns x_0 and y_0 represent the x and y coordinate of the center of the ellipse. Column φ represents the orientations of each ellipse, and lastly column μ represents the gray levels. For preclinical applications, the FBP was run on two realistic projection data sets obtained from commercial CT scanners. We first present numerical simulation results, followed by the preclinical application results.

Table 1. Shepp Logan phantom parameters

No.	a	b	x_0	y_0	φ	μ
1	0.6900	0.900	0	0	0	1.0
2	0.6792	0.882	0	0	0	-0.8
3	0.4100	0.160	-0.22	0	108	-0.2
4	0.3100	0.110	0.22	0	72	-0.2
5	0.2100	0.250	0	0.35	0	0.2
6	0.0460	0.046	0	0.1	0	0.2
7	0.0460	0.023	-0.08	-0.65	0	0.1
8	0.0460	0.023	0.06	-0.65	90	0.1

B. Evaluation Criteria

Prior to running the experiments, criteria for evaluating the performance of algorithms had to be established. To evaluate the quality of the reconstructed images, the Root Mean Square Error(RMSE), Structural SIMilarity (SSIM), and Full Width at Half Maximum(FWHM) were calculated.

RMSE is a traditional measure of image quality based on error sensitivity [11]. It's a measure of how much on average values in a reconstructed image differ from a reference image of perfect quality. The smaller the RMSE is, the better the image quality is. However, the RMSE doesn't account for the perceived quality of the images. Studies have shown that the human visual system (HVS) is highly adapted to extract structural information from the viewing field.

The SSIM measures the similarity between two images [11]. A measure of structural information change can provide a good approximation to perceived image distortion. An SSIM index is a measure of how structurally similar two images are. SSIM requires access to a full reference image to work. 8 by 8 area in the test image is chosen and compared to the corresponding area in the reference image. A local SSIM index value is given based on how similar these two patches are. This process is repeated to obtain local SSIM index values over the entire image. Then, the mean SSIM is calculated [11]. The reason it is broken into small chunks for comparison is that in theory our eyes focus on one small area at a time. The SSIM therefore improves upon RMSE by including human visual perception in assessing image quality. The SSIM index is a value between 0 and 1 inclusive. A value of 1 means two images are fully structurally similar, while a value of 0 means there is no similarity at all [11].

FWHM is the size of the smallest feature that can be detected. It can also be thought of as the smallest distance two objects can be from each other, and still be differentiated. FWHM doesn't require reference images like RMSE and SSIM [12]. Usually a point spread function of a system under study is used to obtain FWHM. Because it is quite difficult to obtain point spread function, we resorted to a method similar to the one used by Schlueter *et al* [12]. To compute FWHM, a point in a homogeneous region (e.g. a small ellipse in the Shepp-Logan phantom) in the reconstructed image was chosen. A line radiating from this point to the edge was sampled to generate a single step function (SSF). Its derivative was calculated via forward approximation to yield the Line spread function (LSF). A Gaussian model was fitted to the LSF. Then, FWHM is calculated from the Gaussian fit. FWHM is the difference between two values of the independent variable at which the value of the Gaussian function is half maximum. There can only be two such points for a Gaussian function. This process was repeated for equiangular lines from the point. Their FWHM was averaged to get the mean FWHM. The smaller the FWHM value is, the better the spatial resolution is. It must be mentioned that this measurement is subjective because a point has to be picked manually from the reconstructed image. The point picked matters a lot in final determination of FWHM value. However, it is fair for all the methods to be evaluated.

C. Numerical Simulations

720 projections were analytically obtained over 2π by using the parameters in Table 1 for the Shepp Logan phantom. The distance D from the x-ray focal spot to the origin of the object was 500 mm, and the radius of the phantom was 100 mm. The projections were spaced 0.5° from each other ($\Delta\beta$). The detector had 600 cells, which implies that each projection contains 600 ray sums through the object. FBP algorithm was applied to the projection data four times, once for each type of the aforementioned derivative computing method. Fig. 2 shows the reconstructed 512x512 images, Fig.3 shows representative profiles along the vertical and horizontal lines in Fig. 2, and Table 2 lists quantitative analysis results.

Table 2. Quantitative analysis of Shepp Logan phantom results

	RMSE	SSIM	FWHM
Forward	0.1197	0.8721	0.0998
Backward	0.1197	0.8721	0.0998
Central	0.1161	0.9062	0.1090
Spline	0.1204	0.8851	0.0981

As one can see it's hard to tell the images one from the other particularly those from FDDA, BDDA and CSDA. However, the RMSE shows the difference. FBP with CDDA yielded the smallest RMSE among the four. FDDA and BDDA were tied for second, while CSDA came in last place. FDDA and BDDA have the same value as they are essentially just shifted versions of each other. It must be noted that these results were expected because appropriate pixel shift correction as stated earlier was applied. RMSE disregards this shift. The error between the reference and images are also shown in Fig. 2. CDDA has the least error of the four, while FDDA, BDDA and CSDA follow closely. Table 2 column 3 lists the SSIM results. CDDA came first however CSDA came in second this time, and FDDA and BDDA both came in last. SSIM index for CDDA is 2.38% higher than second place CSDA, while CSDA was 1.49%

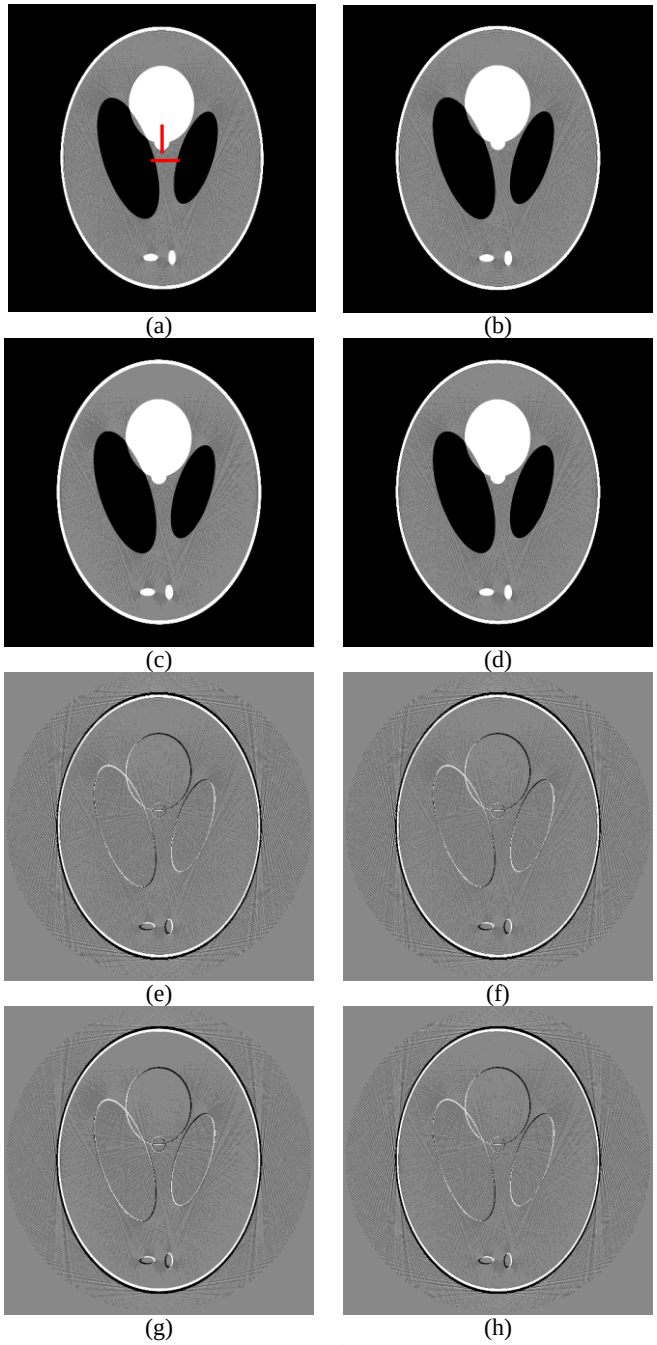


Fig. 2. Numerical simulation results of the Shepp-Logan phantom. (a)-(d) are images reconstructed by the Forward, Backward, Central and Spline methods for derivative, respectively. (e)-(h) are the corresponding error images of (a)-(d) with respect to the ground truth. The display window for (a)-(d) is [0.15,0.25] and the display window for (e)-(h) is [-0.05, 0.05].

higher than FDDA and BDDA. Again, FDDA and BDDA were tied. CDDA method produced the most structurally similar image to the original. Table 2 column 4 lists FWHM results from the Shepp-Logan phantom. This time CSDA performed the best, FDDA and BDDA were tied for second, and surprisingly CDDA is last. The fact that CDDA does an averaging of values around the point of interest to compute the derivative causes it to incur a penalty in spatial resolution. In fact, this is a tradeoff between image noise and spatial resolution. The sharper the image is, the higher the spatial resolution is and the higher the noise level (e.g. RMSE) is.

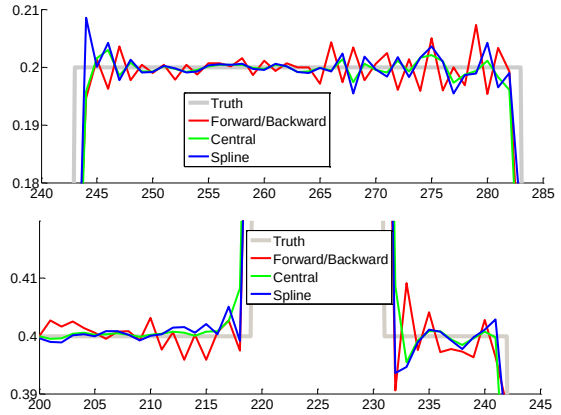


Fig.3. Representative profiles along the horizontal (top) and vertical red lines indicated in Fig. 2 (a).

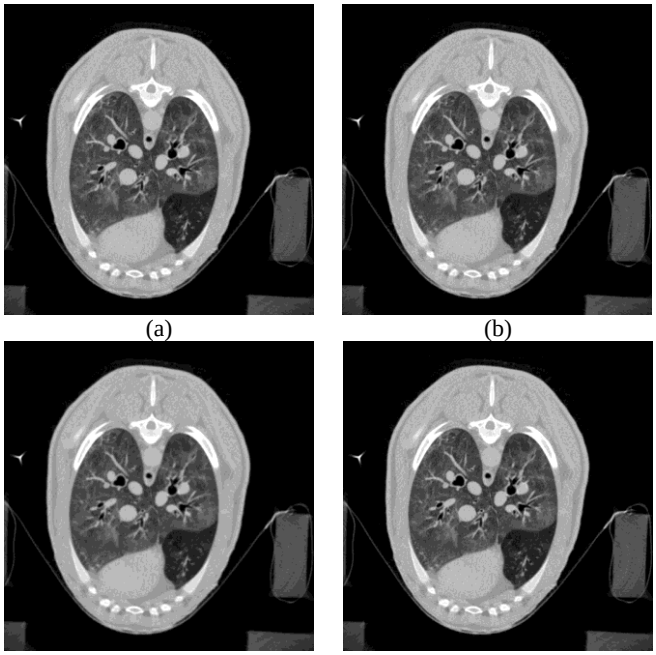
D. Realistic Data Sets

The FBP algorithm run on two different real preclinical data sets. The 1st data set was acquired on a Siemens SOMATOM 64 CT scanner. It consisted of 1160 projections over 2π scanning range of a slice through the chest of a sheep. The projections were spaced 0.31° from each other. The detector had 672 detector cells spaced 0.08°. This data set was a combined 1160x672 array. The distance *D* from the x-ray focal spot to the origin of the object was 570mm. Similar to the numerical simulations, the FBP algorithm was applied to the data four times, and once for each type of derivative computing method. The reconstructed images are shown in Fig. 4. The reconstructions are look generally good and similar. An objective measurement is needed to tell the difference. RMSE and SSIM cannot be relied upon as they require access to a perfect reference image which is impossible for a preclinical application. Here, we just chose to compute spatial resolution. Table 3 column 1 lists the results. The CSDA had the highest spatial resolution. It performed 5.598% better than second place. FDDA and BDDA were tied for second, and CDDA came in close last. Those results are consistent with the aforementioned numerical simulations.

Table 3. FWHM results of real datasets

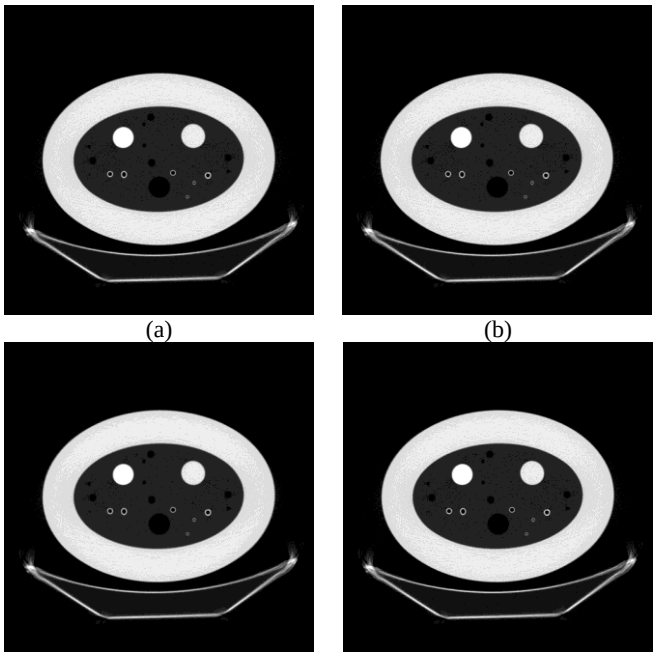
	Real Dataset 1	Real Dataset 2
Forward	1.7025	1.9848
Backward	1.7025	1.9848
Central	1.7035	2.0418
Spline	1.6132	1.8895

The 2nd data set was acquired on a Discovery CT750 HD scanner. It consisted of 984 projections over 2π range for a slice through a physical phantom. The projections were spaced 0.37° from each other. For this kind of GE CT scanner, the detector had 888 detector cells spaced 0.063° . The dataset was a combined 984x888 array. The distance *D* from the x-ray focal spot to the origin of the object was 538.52 mm. Again, the FBP algorithm was applied four



(a) (b) (c) (d)

Fig. 4. Results from the first real dataset. (a)-(d) are reconstructed by the Forward, Backward and Central differences and Spline method, respectively. The display window is [-1000, 500]HU.



(a) (b) (c) (d)

Fig. 5. Results from the second real dataset. (a)-(d) are reconstructed by using Forward, Backward, Central differences and Spline method, respectively. The display window is [-1000, 400]HU.

times to projections, once for each type of derivative computing method. The reconstructed images are shown in Fig. 5. Similar to the first preclinical data set, the reconstructed images are generally good and quite similar. Quantitative analysis results of FWHM are listed in Table 3. The CSDA had the highest spatial resolution. It performed 5% better than the second place FDDA and BDDA methods. CDDA came in last. Again, it confirms that the CSDA method can produce the best spatial resolution.

V. DISCUSSIONS AND CONCLUSION

For the ideal imaging model, there is no noise in the projections. This is not the case in reality. In fact, x-ray imaging equipment generates noise with the production of x-rays [3]. The detected x-ray photons including noise are converted to electrical signal for storage. The more the x-ray photons are generated, the less noise that is superimposed. A stronger x-ray flux may have less noise. However, it delivers more radiation dose. If the x-ray signal is relatively higher than noise, the effect of noise can be disregarded or filtered out. The key here is that the signal strength grows much larger than the noise such that the signal is separable from the noise or the noise is negligible [15]. The parameter referred to here is the signal to noise ratio (SNR) which is the ratio of the strength of signal to noise expressed in decibel. For x-ray tomography a minimum SNR needed to obtain approximately accurate reconstructed images is desired. A higher SNR might mean better results at a cost of higher radiation dose to the patient. To assess the effect of noise as well as measure the sensitivity of the FBP algorithm to it, we added white Gaussian noise to all simulated data with a varying SNR, and calculated RMSE, SSIM, and FWHM. It has been proved that the well-accepted Poisson noise model is equivalent to the Gaussian noise model in the after-log linear integrals. Here the uniform Gaussian noise can be understood as the after-log Poisson noise with an ideal bow-tie filter to make it uniform. The signal to noise ratio was changed from 20 to 140 decibels in 10 increments, and results calculated at each step. Without loss of generality, the CDDA method (the most accurate) was used to evaluate the stability for the derivation computing part. Table 4 lists the RMSE, SSIM and FWHM results of running the FBP algorithm employing CDDA on a simulated noisy data set from the simulation experiment. An SNR of 40 decibels yields results that approximate the noiseless case from Table 2. This means an SNR of or around 40 is the minimum required to obtain good reconstructions. If better results are still required, a bigger SNR has to be risked.

In this paper, we focused on four methods of the numerical derivative in an FBP algorithm. There are many methods that haven't been discussed including other difference and spline variants. It is hoped that this paper may provide motivation for further study of the derivative operation. Another thing that has not been discussed is the computational cost of methods used. The CSDA method is the most computational expensive as it involves more steps and calculations than the rest. The rest have similar if not the same cost. Is the modest increase in spatial resolution worth spline's computational cost? This is something that maybe considered, but given the power of modern computers cost is negligible.

In conclusion, we implemented the equiangular FBP algorithm for fan beam geometry with a focus on derivative

approximation, and numerically tested them using the Shepp-Logan phantom and two preclinical datasets. Our results show that the derivative calculated via central difference approximation produces the least error and most structurally similar reconstructed image, whereas the CSDA produced the highest spatial resolution. This doesn't mean that FDDA and BDDA are bad as all methods performed relatively well. However, if highest spatial resolution or least error is desired then CSDA or CDDA are recommended, respectively. For most cases or when in doubt, the central difference derivative approximation method is recommended for lower noise.

Table 4. Shepp Logan central case RMSE, SSIM, and FWHM results at different signal to noise ratios

SNR	RMSE	SSIM	FWHM(mm)
20	0.7986	0.1945	1.985
30	0.1179	0.6396	1.288
40	0.1162	0.8725	1.192
50	0.1161	0.9029	1.115
60	0.1161	0.9058	1.080
70	0.1161	0.9062	1.094
80	0.1161	0.9062	1.080
90	0.1161	0.9062	1.083
100	0.1161	0.9062	1.087
110	0.1161	0.9062	1.092
120	0.1161	0.9062	1.090
130	0.1161	0.9062	1.090
140	0.1161	0.9062	1.090

ACKNOWLEDGMENT

This work was partially supported by the NSF CAREER Award CBET-1540898.

REFERENCES

[1] A. Kak and M. Slaney. "Principles of Computerized Tomographic Imaging, IEEE press, 1994

[2] L. Shepp and B. Logan. "The Fourier reconstruction of a head section," IEEE Transactions on Nuclear Science, NS-21 (3): 21-43

[3] G. Zeng. Medical image Reconstruction: A conceptual Tutorial, China Higher Education Press, 2009

[4] A. Oppeinheim and R. Schafer. Discrete-time signal processing, Prentice hall, 2010.

[5] R. Lyons. Understanding Digital Processing, Prentice hall, 2010

[6] E. Weisstein. Finite Difference, From Mathworld-A Wolfram Web Resource, https://mathworld.wolfram.com/FiniteDifference.html.

[7] E. Weisstein. Central Difference, From MathWorld-A Wolfram Web Resource, https://mathworld.wolfram.com/CentralDifference.html

[8] M. Abramowitz and I. Stegun. Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables, New York, 1972

[9] C. De Boor. A practical Guide to Splines, Springer-Verlag, 1978

[10] W. Press and S. Teukolsky. Numerical Recipes in C: The Art of Scientific Computing, Cambridge University Press, 1988

[11] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, "Image quality assessment: from error visibility to structural similarity," Image Processing, IEEE Transactions on, vol. 13, pp. 600-612, 2004

[12] F. Schlueter, G. Wang, P. Hsieh, J. Brink, D. Balfe, and M. Vannier, "Longitudinal Image Deblurring in Spiral CT," Medical Physics, 1994

[13] G. Ramachandran and A. Lakshminarayanan. "Three dimensional reconstructions from radiographs and electron micrographs: Application of convolution instead of Fourier transforms," Proc.Nat. Acad. Sci., vol.68, pp. 2236-2240, 1971

[14] E. Weisstein. Cubic Spline. From MathWorld—A Wolfram Web Resource. http://mathworld.wolfram.com/CubicSpline.html

[15] S. Smith. The Scientist and Engineer's Guide to Digital Signal Processing, California Technical Publishing, 1997