A Fast Method to Emulate an Iterative POCS Image Reconstruction Algorithm

Gengsheng L. Zeng

***Abstract*—Iterative image reconstruction algorithms are commonly used to optimize an objective function, especially when the objective function is non-quadratic. Generally speaking, the iterative algorithms are computationally inefficient. This paper presents a fast algorithm that has one backprojection and no forward projection. This fast algorithm can be used to replace many iterative algorithms. This paper derives an *ad hoc* method to solve an optimization problem. The non-quadratic constraint, for example, an edge-preserving de-noising constraint is implemented as a non-linear filter. The algorithm is derived based on the POCS (projections onto projections onto convex sets) approach. A windowed FBP (filtered backprojection) algorithm enforces the data fidelity. An iterative procedure, divided into segments, enforces edge-enhancement denoising. Each segment performs non-linear filtering. The derived iterative algorithm is computationally efficient. It contains only one backprojection and no forward projection. Low-dose CT data are used for algorithm feasibility studies. The non-linearity is implemented as an edge-enhancing noise-smoothing filter. The proposed iterative algorithm is an ad hoc method. The patient studies results demonstrate its effectiveness in processing low-dose x-ray CT data.**

Keywords—iterative image reconstruction; edge-enhancing de-noising; non-linear filter; x-ray CT; fast algorithms

I. Introduction

De-nosing is a classic topic in image processing and image reconstruction. The most traditional approach is to smooth the image with neighboring pixels. This is achieved by image domain convolution or Fourier domain low-pass filtering [1-4]. However, its side effects include blurred boundaries and reduced image contrast.

The filtered backprojection (FBP) is the working horse in x-ray CT industry and in other imaging modalities. In order to reduce reconstruction noise and image artifacts, iterative reconstruction algorithms are commonly used to replace the FBP. Iterative algorithms are used to optimize an objective function. They are versatile in providing useful features, reducing noise, and maintaining the sharp edges. However, iterative algorithms are computationally inefficient and require much more computational resources than the FBP approach.

If the objective function is quadratic, we previously derived an FBP algorithm that is able to give the result of an iterative algorithm in one step, using one backprojection. For an objective function that reduces noise while leaves the object edges unchanged is not likely be quadratic; our one-step FBP algorithm is not effective in this case. As a remedy, one can perform post-filtering to reduce the noise and maintain the edges. Many nonlinear filters can be used to reduce the noise while keeping the edges un-smoothed, such as the Huber filter, median filter, bilateral filter, guided filter, and the like [5-9]. This approach first reconstructs the image, and then applies an edge-preserving post filter.

G. L. Z is with Weber State University/University of Utah. E-mail: larryzeng@weber.edu

An iterative algorithm with embedded nonlinearity is not equivalent to a one-step reconstruction followed by post filtering. The iterative algorithms with some special constraints can outperform the one-step FBP algorithm followed by a post-filter. The problem is that the iterative algorithms require much more computational resources than the one-step FBP algorithm. The goal of this paper is to develop a fast algorithm that has a non-quadratic regularization constraint, such as edge-preserving denoising.

One big difference between an FBP solution and an iterative solution is that the FBP solution does not need any initial conditions, while the iterative solution depends on the initial image. We first derive a windowed FBP algorithm that depends on the initial image. We then apply a non-linear filter (for example, an edge-preserving de-nosing filter) to the windowed FBP result. Next, we use the output of the non-linear filter as the “initial” image for another iteration.

This windowed FBP algorithm is not the conventional FBP algorithm; it is equivalent to *n* iterations of an iterative Landweber algorithm, where *n* can be any positive integer. The non-linear filter can be an edge-preserving noise-smoothing filter (e.g., Huber), median filter, bilateral filter, guided filter, and so on. Instead of “edge-preserving”, we can further make it “edge-enhancing.” Regardless the number of iterations, only one backprojection is needed and no forward projection is required in the proposed algorithm. We will make this point clear next.

II. Methods

A. Algorithm derication

The following gives an overview of the weighted FBP algorithm, which was first introduced in reference [10]. The derivation of the weighted FBP algorithm starts with the iterative Landweber algorithm as presented in first line of (1):

$$\begin{aligned}
&X^{(k)} = X^{(k-1)} + \alpha A^T (P - AX^{(k-1)}) \\
&= \alpha A^T P + (I - \alpha A^T A) X^{(k-1)} \\
&= \alpha A^T P + (I - \alpha A^T A)[\alpha A^T P + (I - \alpha A^T A) X^{(k-2)}] \\
&= \alpha A^T P + (I - \alpha A^T A)\alpha A^T P + (I - \alpha A^T A)^2 X^{(k-2)}
\end{aligned}$$

DOI: 10.12059/Fully3D.2017-11-3108001

$$= \alpha A^T P + (I - \alpha A^T A)\alpha A^T P +$$
$$(I - \alpha A^T A)^2 [\alpha A^T P + (I - \alpha A^T A) X^{(k-3)}]$$
$$= \alpha A^T P + (I - \alpha A^T A)\alpha A^T P +$$
$$(I - \alpha A^T A)^2 \alpha A^T P + (I - \alpha A^T A)^3 X^{(k-3)}$$
$$= \ldots$$
$$= [I + (I - \alpha A^T A) + \ldots +$$
$$(I - \alpha A^T A)^{k-1}]\alpha A^T P + (I - \alpha A^T A)^k X^{(0)}$$
$$= [\sum_{n=0}^{k-1} (I - \alpha A^T A)^n]\alpha A^T P + (I - \alpha A^T A)^k X^{(0)}. \quad (1)$$

This iterative algorithm is used to solve for the system $AX = P$, where X is the image and P is the projection sinogram. The last line of (1) has two terms: the first term depending on the projection sinogram P and the second term depending on the initial image $X^{(0)}$. In (1), the parameter α is the step size and is upper-bounded by the sigular value of the matrix A.

Using Eq. (5) in reference [10], we can further have

$$X^{(k)} = (A^T A)^{-1}[I - (I - \alpha A^T A)^k] A^T P +$$
$$(I - \alpha A^T A)^k X^{(0)} = F^{(k)} P + H^{(k)} X^{(0)} \quad (2)$$

where we define

$$F^{(k)} \equiv (A^T A)^{-1}[I - (I - \alpha A^T A)^k] A^T \text{ and } H^{(k)} \equiv (I - \alpha A^T A)^k. \quad (3)$$

In [10], the initial image $X^{(0)}$ was assumed to me zero, and the first term can be re-written as a modified FBP algorithm. This weighted FBP algorithm is equivalent to k steps of the associated iterative Landweber algorithm. This paper does not assume $X^{(0)} = 0$.

The second term $H^{(k)} X^{(0)}$ in (2) was not discussed in [10] but is considered in this paper. This second term can be considered as an operator H acting on the initial image $X^{(0)}$. The operator H in the matrix form is defined as $H^{(k)} \equiv (I - \alpha A^T A)^k$ as shown in (3). Since A is the projector and A^T is the backprojector, the combined operator $A^T A$ has a point spread function of $1/r$, where r is the distance from the point source. The 2D Fourier transform of $1/r$ is $1/\|\vec{\omega}\|$, where $\vec{\omega}$ is the frequency vector in the Fourier domain. Thus $1/\|\vec{\omega}\|$ is a lowpass filter and $(1 - \alpha/\|\vec{\omega}\|)$ is a highpass filter. Consequently, the Fourier domain representation of H, $(1 - \alpha/\|\vec{\omega}\|)^k$, is also a highpass filter. In computer implementation, the frequency vector $\vec{\omega}$ is discretized, and the parameter α is chosen such that $|(1 - \alpha/\|\vec{\omega}\|)| < 1$ to prevent divergence with a large k.

On the other hand, the first term in (2) can be considered as an operator F acting on the projections P. The operator F in the matrix form is defined as $F^{(k)} \equiv (A^T A)^{-1}[I - (I - \alpha A^T A)^k] A^T$ as shown in (3). This operator consists of three parts: A^T is a backprojector, $I - (I - \alpha A^T A)^k$ is a lowpass window function, and $(A^T A)^{-1}$ is the 2D ramp filter $\|\vec{\omega}\|$.

It is ready to see that $(A^T A)^{-1}$ is the 2D ramp filter $\|\vec{\omega}\|$ because $A^T A$ corresponds to $1/\|\vec{\omega}\|$, and $I - (I - \alpha A^T A)^k$ is a lowpass window function because it corresponds to $1 - (1 - \alpha/\|\vec{\omega}\|)^k$ and $(1 - \alpha/\|\vec{\omega}\|)$ is a highpass filter. If one uses the central-slice theorem, this "backprojection first, then filter" operation F can be equivalently achieved by "filter first, then backproject," which is the weighted FBP procedure. In the weighted FBP procedure a 1D ramp filter $|\omega|$ is used and the lowpass window function is $1 - (1 - \alpha/|\omega|)^k$.

If one wishes, the projection noise weighting can be embedded in the first term. The noise weighting is achieved by modifying the "step size" α as $\alpha = \alpha_0 \times w$ in the window function $I - (I - \alpha A^T A)^k$, where α_0 is a default "step size" and w is the weighting function. A smaller weighting function w for less noisy projections and a larger weighting function w for noisier projections [11, 12]. In this paper, for the sake of simplicity, the noise weighting is not included in algorithm (2).

The above linear algorithm enforces projection data fidelity. Edge-preserved denoising will be enforced by a non-linear filter. The structure of our approach is in the form of "alternating projection," which is sometimes referred to as the POCS ((projections onto projections onto convex sets) method. In other words, it is equivalent to k iterations of the Landweber followed by non-linear edge-preserving filter, then another k iterations of the Landweber followed by non-linear edge-preserving filter, and so on. The "k iterations of the Landweber" is actually achieved by a weighted FBP. This iterative scheme can be made even efficient as follows.

Any filter can be used as the nonlinear filter; for example, an edge-preserving denoising filter can be chosen. Let us symbolically represent the chosen nonlinear filter as G. Thus, the proposed iterative algorithm can be expressed as

$$Y^{(1)} = G[F^{(k)} P + 0],$$
$$Y^{(2)} = G[F^{(k)} P + H^{(k)} Y^{(1)}],$$
$$Y^{(3)} = G[F^{(k)} P + H^{(k)} Y^{(2)}],$$
$$\ldots$$
$$Y^{(n)} = G[F^{(k)} P + H^{(k)} Y^{(n-1)}]. \quad (4)$$

The final result for the reconstructed image X is $Y^{(n)}$ with a "segment" number n. Each segment contains k "iterations" as indicated in the superscript of $F^{(k)}$ if $F^{(k)}$ were implemented as an iterative algorithm. In our algorithm both $F^{(k)}$ and $H^{(k)}$ are one-step procedures, and k is just a parameter.

We use notation $Y^{(n)}$ (instead of $X^{(n)}$) is to avoid the confusion with the result of the iterative algorithm (2) which does not use any nonlinear filters. In other words, notation $X^{(k)}$ is for the results of the iterative algorithm (2) without any nonlinearity involved; notation $Y^{(n)}$ is for the results of the iterative algorithm (4) with a nonlinear filter G.

B. Implementation considerations

The implementation of the first term in (2), $F^{(k)}P$, has been explained in details in [10] as an FBP procedure. The first step in this modified FBP algorithm is one-dimensional windowed ramp filtering of the sinogram data. The Fourier domain window function is given as

$$window(\omega) = 1 - (1 - \frac{\alpha}{|\omega|})^k \text{, when } \omega \neq 0 \text{; } window(0) = 1 \text{.} \quad (5)$$

Here ω is the frequency and $\alpha > 0$ is the "step size." If noise weighting is to be incorporated, α is a function of the reciprocal of the noise variance [11, 12]. The second step is to perform backprojection as in a usual FBP algorithm.

When this first step is implemented in the Fourier domain, the frequency ω is discretized in the range of [-0.5, 0.5]. In order to reduce the bias errors, the sinogram P needs to be zero-padded along the detector dimension, say, to reach the array size of 2048 [14]. The selection of the parameter α depends on the frequency sampling interval $\Delta\omega$ so that

$$|1 - \alpha / \Delta\omega| < 1 \text{.} \quad (6)$$

For example, if the zero-padded 1D array size is 2048, which corresponds to $\Delta\omega = 1/2048$. In this case,

$$0 < \alpha < 2\Delta\omega = 1/1024 \text{ .} \quad (7)$$

The second term in (4) $H^{(k)}Y^{(m)}$ is a high-pass filtering of the image $Y^{(m)}$. The implementation of it is first to evaluate the two-dimensional (2D) Fourier transform of the image $Y^{(m)}$, and then to multiply the transformed image by

$$(1 - \alpha / \|\vec{\omega}\|)^k \quad (8)$$

and finally to calculate the inverse 2D Fourier transform. Neither projection nor backprojection is required for the implementation of the second term in (4).

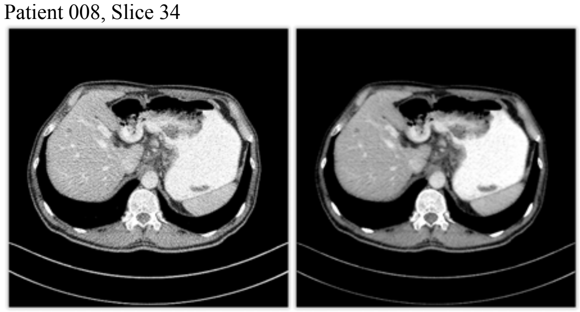
III. Results

This algorithm was developed while the author was participating the 2016 Low-Dose CT Grand Challenge, organized by NIH (National Institutes of Health), AAPM (America Association of Physicists in Medicine) and Mayo clinic. The author was the 3rd place. Dr. Cynthia McCollough, the main organizer of the Grand Challenge from Mayo clinic, provided us some low-dose x-ray CT patient torso images. The image slice was 512 x 512 and the slice thickness was 3 mm. The image volumes were first forward projected using the parallel-beam geometry, to generate projection sinograms. The parallel-beam sinograms are then used as the inputs for our proposed algorithm. It is well understood that the re-projected sinogram does not carry exactly the same information as the original sinogram. The spatial resolution may get degraded. The original projection data generally do not satisfy the data consistency conditions, while the re-projected data always satisfy the data consistency conditions. The noise in the original data can be assumed to be independent; however, the noise in the re-projected data is not independent.

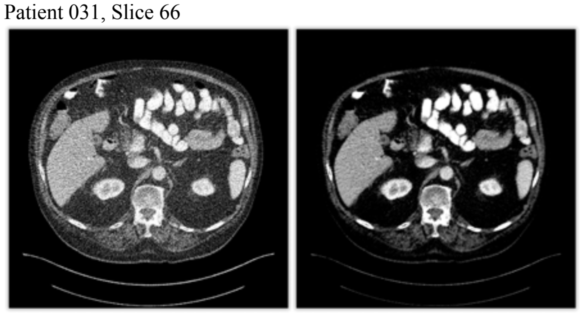
We used the following parameters for image processing: α = 0.001, k = 2000, local 3D region size = 3 x 3 x 3, ε = 0.0005, β = 1.05, and n = 10. The second equation in (13) was used instead of Eq. (10). Four pairs of images are shown in Fig. 1. Each pair is one slice from a patient. All images are shown in the Hounsfield Unit window of [900, 1200]. Images on the left are reconstructed via the conventional FBP algorithm provided by the Grand Challenge organizer. Images on the right are reconstructed with the proposed algorithm. The image reconstruction algorithms were programmed in MATLAB and were run on a CPU in a Linux operation system. The reconstruction time for the conventional FBP was 3.59 seconds per slice; for the proposed algorithm was 3.72 seconds per slice.

Figure 2 provides two additional reconstructed images for Patient #136 (slice 30). Fig. 2 (Left) is the reconstruction with windowed FBP (using k = 2000) and no nonlinear filter is applied. Fig. 2 (Right) shows the reconstruction using proposed algorithm (using $k = \infty$, that is, a conventional FBP is used as the initial image). From this example, it seems that the nonlinear filter plays a more important role than the window function in the weighted FBP algorithm. However, the window function will make significant difference when the projection noise weighting is included and severe streaking artifacts are present if no noise weighting is used, as shown in [11].

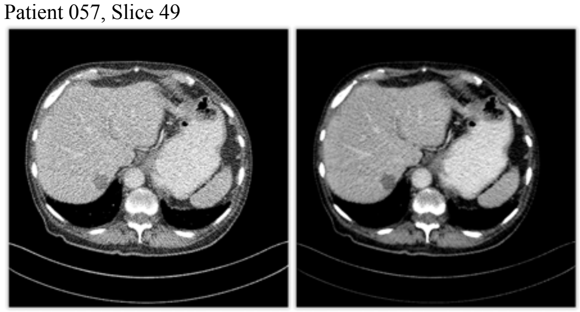
Patient 008, Slice 34



Patient 031, Slice 66



Patient 057, Slice 49



Patient 136, Slice 30

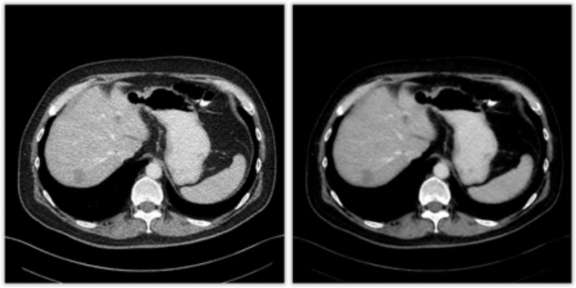


Figure 1. Image reconstruction results for 7 patients. Left: conventional FBP. Right: proposed algorithm.

Patient 136, Slice 30

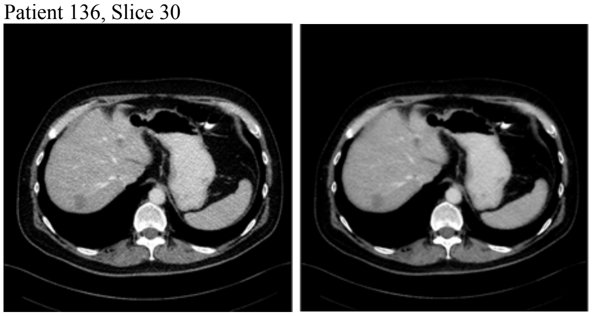


Figure 2. Reconstructed image for Patient136 (slice 30). Left: windowed FBP reconstuction without nonlinear filtering. Right: proposed algorithm with $k = \infty$ (i.e., conventinal FBP with nonlinear filtering).

IV. Discussion and Conclusions

The conventional iterative algorithm consists of projection, backprojection, and procedures corresponding to special constraints. This paper has presented the following concepts and methods:

An iterative algorithm can have a structure of “segments.” Each segment can contain multiple (k) linear iterations. At the end of each segment, a non-linear is performed.

There are k linear iterations for every segment, and these k linear iterations can be implemented in one step, in the form of an FBP algorithm. Compared with a conventional FBP algorithm, this modified FBP algorithm has an extra window function applied to the ramp filter. This window function depends on the iteration number k and the step size α. It can also depend on the noise weighting model. The fast algorithm takes the advantage that the modified FBP has two parts: one part depends only on the projection sinogram, and the other part depends only on the initial image. The part depending on the projection sinogram can be pre-calculated. The other part is a high-pass filtered version of the initial image for the current segment. Neither projector nor backprojector is needed in this highpass filter. This highpass filter can be implemented as: performing the 2D FFT on the initial image, multiplication of the transfer function (8), and performing the 2D IFFT (inverse fast Fourier transform). The initial image is the result from the previous segment.

In our application, the weighted FBP $F^{(k)}$ and the highpass filter $H^{(k)}$ (8) are 2D; the edge-enhancing denoising filter is 3D. The users are free to change them according to their applications. A fast algorithm relies on the balance of the k (the number of linear iterations, which are actually accompished by a pre-calculated weighed FBP image) and n (the number of segments). One can choose a large k (say, 1000) and small n (say, 5), depending on the applications and tasks. When $n = 1$, the proposed algorithm reduces to the reconstruction plus post-filtering algorithm.

We have applied the proposed iterative algorithm to some patient low-dose x-ray CT data for the feasibility study. One can clearly observe the image quality improvement over the conventional FBP method. The computation time is slightly longer than that of the conventional FBP algorithm.

V. References

[1] L. G. Shapiro and G. C. Stockman, *Computer Vision*, (Prentice Hall, 2001).

[2] M. S. Nixon and A. S. Aguado, *Feature Extraction and Image Processing*, (Academic Press, 2008).

[3] R. C. Gonzalez and R. E. Woods, *Digital Image Processing,* (Prentice Hall, 2008).

[4] C. J. Solomon and T. P. Breckon, *Fundamentals of Digital Image Processing: A Practical Approach with Examples in Matlab*, (Wiley-Blackwell, 2010).

[5] P. J. Huber, *Robust Statistics*, (New York: Wiley, 1981).

[6] R. L. Stevenson, B. E. Schmitz, and E. J. Delp, “Discontinuity pre-serving regularization of inverse visual problems,” *IEEE Trans. Syst., Man, Cybern*., vol. 24, no. 3, pp. 455–469 (1994).

[7] T. Huang, G. Yang, and G. Tang, “A fast two-dimensional median filtering algorithm,” *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 27, no. 1, pp. 13–18 (1979).

[8] C. Tomasi and R. Manduchi, “Bilateral Filtering for Gray and Color Images,” in *Proceedings of the 1998 IEEE International Conference on Computer Vision* (IEEE, Bombay, India, 1998).

[9] K. He, J. Sun, and X. Tang, "Guided Image Filtering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, Issue 6, pp. 1397-1409 (2013).

[10] G. L. Zeng, “A filtered backprojection algorithm with characteristics of the iterative Landweber algorithm,” *Med. Phys*., vol. 39, pp. 603-607 (2012).

[11] G. L. Zeng and A. Zamyatin, “A filtered backprojection algorithm with ray-by-ray noise weighting,” *Med. Phys*. vol. 40, 031113; http://dx.doi.org/10.1118/1.4790696 (2013).

[12] G. L. Zeng, Y. Li and A. Zamyatin, “Iterative total-variation reconstruction vs. weighted filtered-backprojection reconstruction with edge-preserving filtering,” *Phys. Med. Biol*. vol. 58, pp. 3413-3431 (2013).

[13] K. He and J. Sun, “Fast guided filter,” Computer *Vision and Pattern Recognition (cs.CV),* http://arxiv.org/abs/1505.00996

[14] G. L. Zeng, “Re-visit of the ramp filter,” *IEEE Trans. Nucl. Sci*., vol. 62, no. 1, pp.131-136 (2015).