Iterative circular conebeam CT reconstruction using fast hierarchical backprojection/reprojection operators

Jeffrey Brokish[a], Daniel B. Keesing[a], Yoram Bresler[ab]
[a]InstaRecon, Inc., Champaign, IL USA 61820
[b]Dept. of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign

ABSTRACT

This is the first report on a new fast statistical iterative reconstruction algorithm for conebeam with a circular source trajectory, accelerated by InstaRecon's fast $O(N^3\log N)$ hierarchical cone beam backprojection[1] and reprojection algorithms. We report on the results of image quality and run-time comparisons with iterative algorithms based on conventional backprojection and reprojection. We demonstrate that the iterative algorithm introduced here can provide image quality indistinguishable from an iterative algorithm using conventional BP/RP operators, while providing almost a 10x speedup in reconstruction rates. Combining the 10x algorithmic acceleration with additional hardware acceleration by FPGA, Cell, or GPU implementation, this work indicates the feasibility of iterative reconstruction algorithms for dose reduction and image quality improvement in routine CT practice, at competitive speeds and affordable cost.

Keywords: Tomography, Cone-Beam, Iterative Reconstruction, Hierarchical Algorithm, Fast Algorithm

1. INTRODUCTION

Incorporating statistical and physical models of the data acquisition, iterative CT reconstruction algorithms reduce noise and artifacts in images, which has implications for reduced dose imaging. However, iterative algorithms are computationally expensive, as each iteration requires a forward and backward projection operation. These backprojection/reprojection (BP/RP) operators have computational complexity of $O(N^4)$ for 3D cone-beam reconstruction of an $N \times N \times N$ volume. They are therefore the computational bottleneck of iterative reconstruction, just as the BP operation is the bottleneck of standard filtered backprojection. There exist algorithms[1] that reduce the algorithmic complexity of these operators to $O(N^3\log N)$, offering the potential for tremendous speedups. Here, we apply InstaRecon's fast hierarchical BP/RP operators to accelerate an iterative algorithm for the 3D circular cone-beam geometry.

We report on the results of image quality and run-time comparisons with iterative algorithms based on conventional backprojection and reprojection. We demonstrate that the iterative algorithm introduced here can provide image quality indistinguishable from an iterative algorithm using conventional BP/RP operators, while providing almost a 10x speedup in reconstruction rates. Combining the 10x algorithmic acceleration with additional hardware acceleration by FPGA, Cell, or GPU implementation, this work indicates the feasibility of iterative reconstruction algorithms for dose reduction and image quality improvement in routine CT practice, at competitive speeds and affordable cost.

The circular cone beam geometry considered here uses a curved detector panel. For simplicity, the detector panel is logically translated to the center of rotation. The source orbits on a circular trajectory in the x-y plane at a distance D from the center of rotation. Projection data is sampled at P evenly spaced positions along this trajectory. The projection data $g[u, v, p]$ is indexed by detector channel u, row v, and source position p.

2. ITERATIVE ALGORITHM DESCRIPTION

There are various methods for implementing an iterative reconstruction scheme. The choice of a cost function includes weighted least squares, or Poisson or other likelihood, and regularizers such as smoothness penalties or priors. The optimization strategy might entail conjugate-gradients (CG), ordered subsets, or iterative coordinate descent. For this evaluation we chose a penalized weighted least squares (PWLS) cost function of the form

$$J(f) = \|y - Rf\|_W^2 + \beta C(f) \tag{1}$$

where *f* is the current estimate of the image, *y* is the measured projection data, *R* is the forward projection operator, *C*(*f*) is a non-quadratic regularization term that favors smooth regions in *f* while preserving sharp transitions, and the coefficient β determines relative weight of the regularization term to the data-match term. The norm above is a 2-norm with weighting *W*, used to model the noise statistics in the measured projections[2]. The standard form of the 'roughness penalty' *C*(*f*) involves a convex penalty function applied to differences between neighboring pixel values. In our formulation, we consider the local neighborhood of the 26 adjacent voxels. We used the Huber function as the penalty function, defined as

$$\phi(t) = \begin{matrix} 0.5t^2 & |t| \leq \delta \\ \delta\,|t| - 0.5\delta^2 & |t| > \delta \end{matrix} \tag{2}$$

This penalty provides an adjustable threshold δ that can be tuned to the level of noise present in an image. Also, the Huber function, along with its first and second derivatives, are easy to calculate.

The minimization strategy chosen was based on conjugate gradients, because this was determined[3] to be among the fastest minimization strategies for cost functions of the form in (1), and the fast hierarchical BP/RP operators are most readily applicable to it. The CG iterations involve calculations of the gradient of (1),

$$\nabla J(f) = R^{*}W(Rf - y) + \beta C'(f) \tag{3}$$

where R^{*} is the backprojection operator, and $C'(f)$ is the gradient of the roughness penalty. As noted in the introduction, the form of (3) indicates that a reprojection and backprojection operator must be applied in each iteration.

3. REPROJECTION OPERATOR

The reprojection operator involves calculating cone-beam projection data from a discretized image volume. The method used in the reprojection operation determines the quality of the estimated projection data, in terms of the ability of the method to accurately represent the underlying continuous object and to produce projection data that is free of aliasing artifacts. Furthermore, the complexity of the technique employed can have a negative impact on the computational efficiency of applying the operator, ultimately increasing the runtime of the iterative algorithm.

We also found the anti-aliasing aspects of the reprojection operator to be of importance when accelerating the reprojection with a hierarchical algorithm. For this work, we incorporate and extend the ideas of the integrated Joseph footprint[4]. For 2D parallel beam geometries, the projection of a single pixel has the form

$$g(\theta,t) = \begin{cases} \dfrac{1}{|\cos\theta|}\Lambda\left(\dfrac{t}{|\cos\theta|}\right) & |\cos\theta| \geq |\sin\theta| \\ \dfrac{1}{|\sin\theta|}\Lambda\left(\dfrac{t}{|\sin\theta|}\right) & |\sin\theta| > |\cos\theta| \end{cases} \tag{4}$$

where $\Lambda(t)$ is the triangle function. Such a footprint has been demonstrated to have no DC aliasing. However, the triangular shape itself does introduce some false high frequency effects. Integration of the footprint over the detector width is typically used to suppress these effects. This also provides better system modeling, where the entire surface of detector is sensitive, acquiring strip integrals of the volume as opposed to line integrals. For more complete control over aliasing effects, we implemented more complex footprint functions. In particular, we extend the footprint to higher order splines. While other choices of footprint are possible with our framework, spline footprints retain the benefits of the Joseph operator, namely the lack of DC aliasing. However, the increased smoothness of higher order splines reduces the false high frequencies that were introduced by the triangle function.

For computational efficiency, the footprint integration is calculated not on the detector panel, but rather detector boundaries are brought to image rows or columns and integration is performed in the image domain. The choice between rows or columns is based on which direction is most orthogonal to the line between the source and center of rotation. Owing to the inverse cosine or sine stretching of the basis function in (4), when the footprint is brought to the image domain it reverts to the standard, non-warped function $\Lambda(x)$. This is an important property: it simplifies the projection calculation, suggests efficient implementations through processing along rows and columns, and, in the case of non-interpolating footprint representations such as splines, indicates that using such footprints require only basis transformations along rows and columns of the image.

The final algorithmic modification in achieving a computationally efficient reprojector is to not calculate integrals on the interval of the detector directly, but rather determine the cumulative integral of the image elements up to each detector boundary. For each detector edge e_i we calculate

$$G(e_i) = \int_{-\infty}^{e_i} \sum_x a_x \Lambda (t - x) dx \tag{5}$$

where a_x are the image values appropriate for representation with footprint $\Lambda(x)$. The cost of (5) is reduced by splitting the integral into two parts

$$G(e_i) = \int_{-\infty}^{\lfloor e_i \rfloor} \sum_x a_x \Lambda (t - x) dx + \int_{\lfloor e_i \rfloor}^{e_i} \sum_x a_x \Lambda (t - x) dx \tag{6}$$

The first integral in (6) is easily determined by a simple discrete index FIR filtering operation on the image coefficients, the results of which can be shared among all such edge-integration calculations. The second integral in (6) involves only those elements whose footprint intersects the edge. The contribution to the interval integration of detector d_i is simply the difference

$$d_i = G(e_{i+1}) - G(e_i) \tag{7}$$

In the case of divergent-beam geometries, such as the cone-beam case we address here, the detector contribution must be weighted with a 1/cos(α), where α indicates the angle the ray makes with the current image axis.

Extending these ideas to the 3D case, x-z or y-z image slabs are projected onto the detector panel. For simplicity in the projection calculation, we assume a separable 2D footprint in each image slab as $\Lambda(x)\Lambda(z)$. This allows us to efficiently compute the 2D footprint integration, as well as providing the ability to select different footprints for x/y and z directions.

4. FAST HIERARCHICAL REPROJECTION

The hierarchical reprojection algorithm is based on two main concepts. The first concept is "divide and conquer," in which the volume is successively divided into smaller non-overlapping volumes, and the reprojection operation is applied to these subvolumes. For CCB with small to moderate cone angles, a pillar-style decomposition is used, as shown in Figure 1. By itself, this decomposition scheme does not yield any reduction in operation counts.

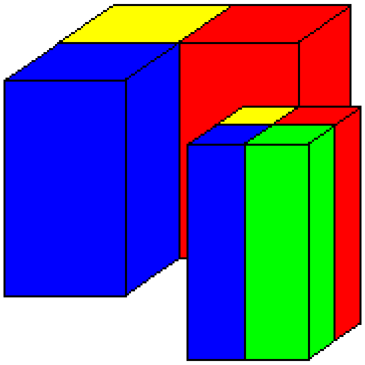

Figure 1. Pillar-Style hierarchical decomposition

The second concept invokes the sampling conditions for CCB projection data[5], where the number of views required to accurately sample the projections of a bandlimited subvolume at the center of the source of rotation is proportional to the size of the subvolume. For the projections of a half-size subvolume, the projection data set can initially be reprojected at a sparser set of $P/2$ projections and then angularly interpolated by a factor of 2, yielding projection data with comparable quality.

This property is extended to apply to subpillars located at any position in the image. When reprojecting a subpillar, the coordinate system is adjusted in u so that the center point of the subpillar $\underline{x}_c$ projects onto the central detector channel. The projection data g_s generated for that subpillar will be a shifted version of the projection data g in the native coordinate system. This can be expressed as

$$g_s[u,v,p] = g[u - u(\underline{x}_c,p),v,p] \qquad (8)$$

where $u(\underline{x}_c, p)$ is the detector channel that point $\underline{x}_c$ projects to when the source is at position p. Introducing this shift in the projection data reduces its angular bandwidth, allowing a sampling argument to be invoked. The centered projections may be reprojected at a sparser set of views and interpolated in source position p without information loss. Since the subpillar is half the size of the original volume, the projection data can be sampled at half the rate, allowing the reprojection operation to be calculated with half the computational effort. After the data has been angularly interpolated, the projections are then "de-centered" by shifting the data in u back to their correct positions.

Figure 2 shows an example of this operation in 2D. First the image is broken down into four quadrants, and each is reprojected at $P/2$ views under a shifted coordinate system relative to the subimage center. The projection data is then interpolated to P views and shifted back to the native coordinate system, indicated by the Interpolate/Shift (I/S) block.

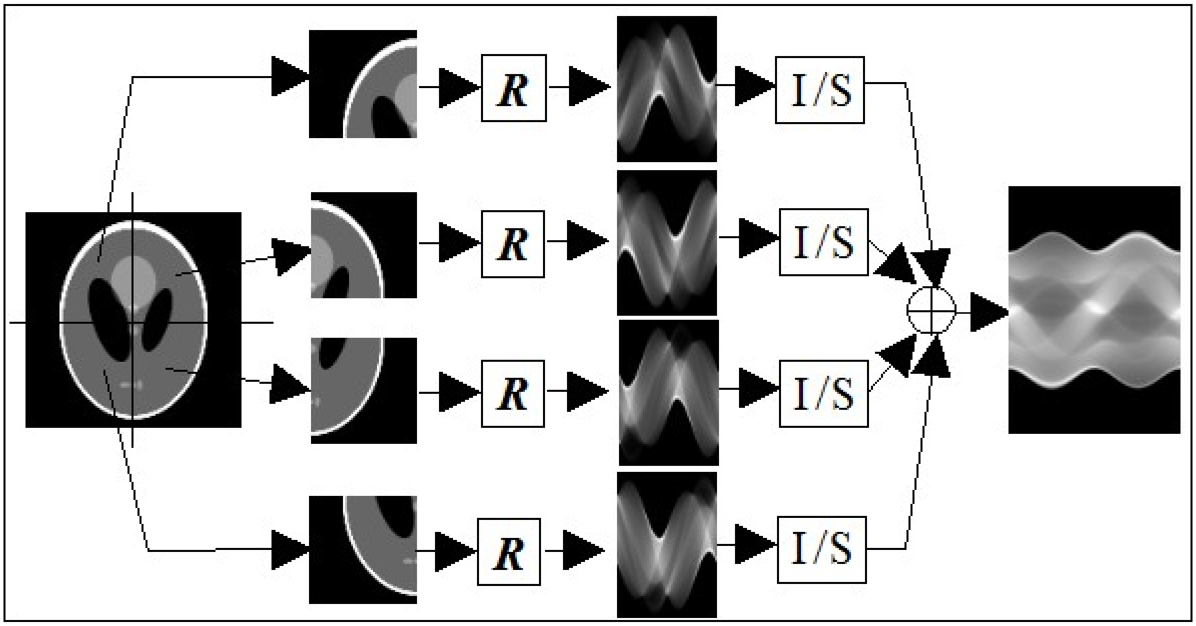

Figure 2. Example of hierarchical reprojection for one level

Applying this process to each subpillar in the volume reduces the computational cost of the reprojection by a factor of two over reprojection of the full volume. The full hierarchical algorithm is formed through recursive application of this process to the reprojection of each subpillar. As shown in Figure 1, a subpillar can be further broken down into smaller subpillars, reducing the number of views required for accurate reprojection by two each time. Applying this at most $\log_2 N$ times yields pillars of a single voxel in cross-section, which will be reprojected at P/N views. When performing the I/S operation, the shift to be performed is determined by the coordinate system relative to the center of the "parent" subpillar. If the subpillar is the top-most subvolume in the decomposition, then it is shifted back to the native coordinate system.

After the $\log_2 N$ decompositions, the reprojection operation uses only P/N projections, reducing the computational complexity of reprojection to $O(N^3)$. The application of the interpolation and shifting procedure to each subpillar in the volume can be shown to be $O(N^3)$ as well. Adding all the work over $\log_2 N$ decomposition stages yields a total complexity of the hierarchical algorithm of $O(N^3 \log_2 N)$.

The fast hierarchical backprojector is constructed as the adjoint of this operation, with the appropriate flow reversal of each step in the algorithm. It is analogous in form to the fast hierarchical backprojection algorithms[1] that have been developed for filtered backprojection reconstruction (FBP). An FBP reconstruction is typically used to form the initial guess in the iterative algorithm, so this step can also be accelerated by hierarchical methods.

There are other fast O(N^3logN) algorithms, but they differ in image quality and speed. An inherent advantage of the fast hierarchical algorithms is that they operate in the native conebeam geometry, avoiding the computational expense of rebinning to a parallel geometry or making approximations that can compromise image quality. Additionally, although other fast BP/RP algorithms scale similarly as O(N^3logN), the constants involved in the hierarchical algorithms are better, such that in practice the hierarchical algorithms are substantially faster than other approaches.

5. DATA AND RESULTS

Projection data was simulated for the Forbild thorax phantom[6] with parameters typical of a 64-row CT scanner. The acquisition and reconstruction configuration parameters are listed in Table 1. Note that the detector spacing values are after the detector panel has been translated to the center of rotation.

Table 1. System geometry

Acquisition		Reconstruction	
Source to object distance	600 mm	Cross-section size	512 x 512
Number of detector channels	983	Voxel size	0.7812 mm
Detector channel spacing	10^{-3} rad	Number of slices	53
Number of detector rows	64		
Row spacing	1 mm		
Number of views	1024		

First, we examine the performance of the hierarchical reprojector. A rasterized version of the thorax phantom was created from the phantom definition and then reprojected using both the conventional and hierarchical reprojectors. Figure 3 shows projection data for the entirety of row 40 for the two reprojectors and, for qualitative comparison, an analytical reprojection of the mathematical definition of the phantom. A cut through the resulting projection data is shown in Figure 4, with the two reprojectors in very good agreement.

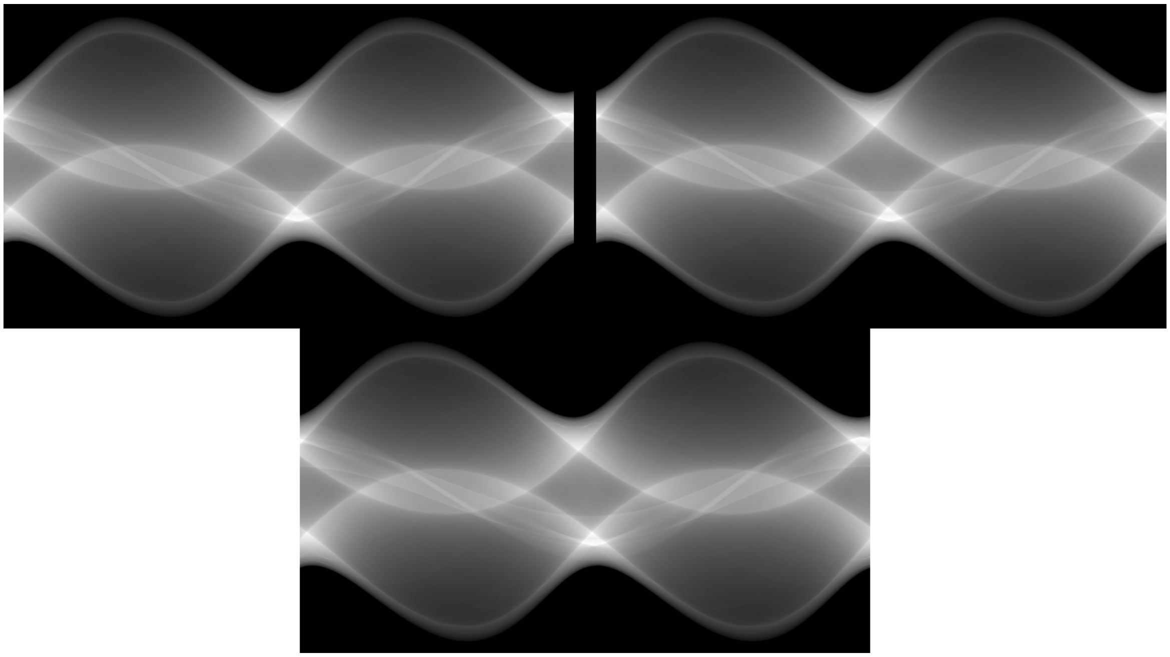

Figure 3. Projection data for row 40 of the conventional reprojector (upper left), the hierarchical reprojector (upper right), and an analytical reprojection of the mathematical phantom (bottom, middle).

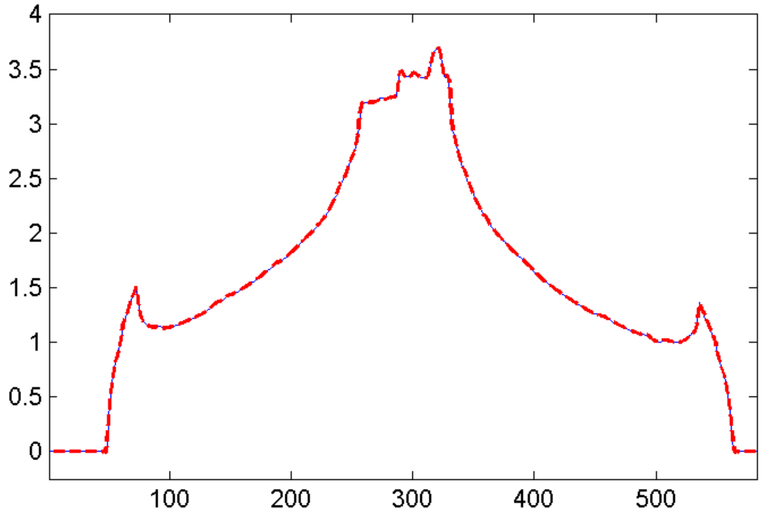

Figure 4. Slice through row 40, view 300 for conventional reprojector (thin, solid, blue) and hierarchical reprojector (thick, dashed, red)

To finish validation of the hierarchical operators, we examine their performance in the iterative algorithm, where we demonstrate that any minor differences between the conventional and hierarchical operators do not manifest as more significant when placed in the iterative loop. For this test, Poisson distributed noise was added to the analytic projection data using incident intensity I_o of 10^5. The same noisy projection data was used in performing an iterative reconstruction with either conventional or hierarchical BP/RP operators. Each algorithm was initialized with a conventional FBP or hierarchical FBP reconstruction, respectively, and then run for 20 iterations. Figure 5 plots the objective function $J(f)$ from (1) for each algorithm, showing extremely similar values at each iteration. Cuts through the final reconstruction after 20 iterations are shown in Figure 6, also demonstrating excellent agreement.

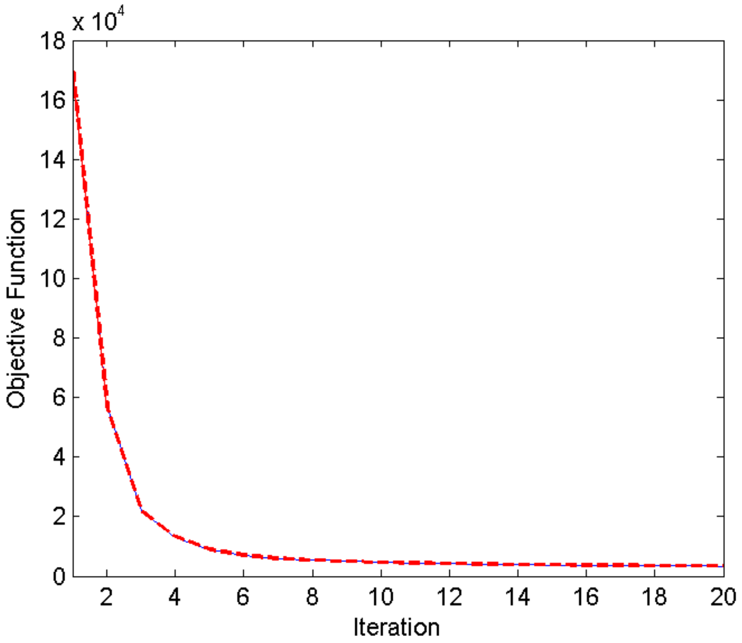

Figure 5. Evolution of objective function with iteration for conventional operator-based iterative algorithm (thin, solid, blue) and hierarchical operator-based iterative algorithm (thick, dashed, red)

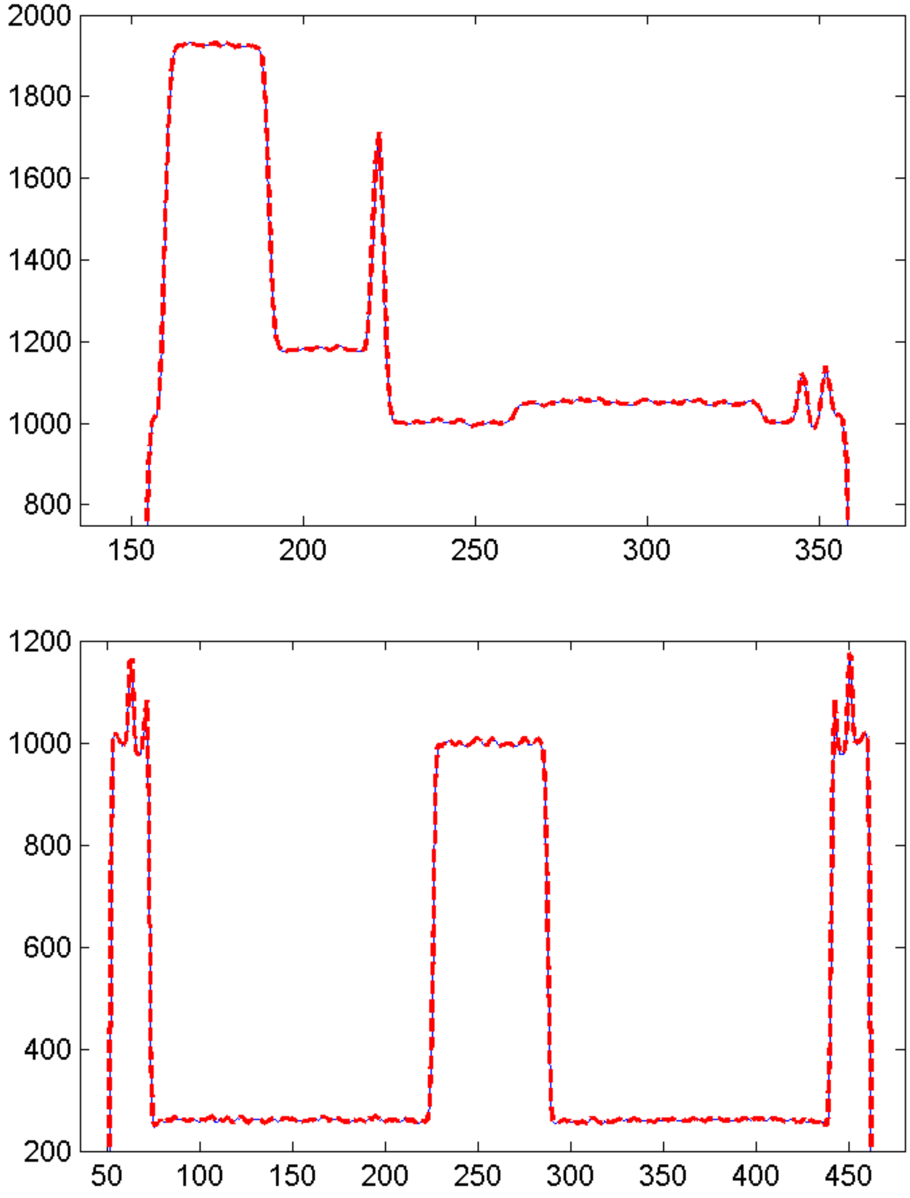

Figure 6. Cuts through reconstruction after 20 iterations for conventional (thin, solid, blue) and hierarchical operator-based iterative algorithm (thick, dashed, red).

Table 2 lists runtime results comparing the iterative algorithm using conventional BP/RP operators to one using the hierarchical BP/RP operators. The reconstruction was run for 10 iterations. The runtime is broken down into the time spent performing the BP/RP operations (the primary computational cost), the regularization penalty term, and other processing, namely the other elements in the CG computational framework such as enforcement of conjugacy in search directions. While the other computational tasks of the iterative algorithm are small relative to the application of the BP/RP operators, they become a more substantial percentage of time once the hierarchical operators have been introduced. Efficient implementations of these calculations is important to preserve a significant net acceleration to the algorithm. The runtimes collected were for single-threaded versions of each algorithm. We are currently working on well-tuned multithreaded implementations to take advantage of modern multiprocessor/multicore systems.

Table 2. Algorithm runtime comparison (times are in seconds)

	Conventional BP/RP	**Hierarchical BP/RP**
BP/RP Operators	6916	649
Regularization Penalty Term	77	77
Other Processing	16	16
Total Runtime	7009	741
BP/RP Operator Speedup		**10.7x**
Total System Speedup		**9.5x**

For a final example, we demonstrate the iterative algorithm's ability to reduce image noise while maintaining image resolution. The data used here is from a Micro-CT scanner. The parameters of the iterative algorithm are chosen to maintain the resolution of the FBP reconstruction for the higher contrast features of the object while performing smoothing in other regions. Slices through the reconstruction are shown in Figure 7, and Figure 8 demonstrates quantitatively the preservation of resolution (edges) while reducing image noise. In this case the standard deviation of the noise was reduced from 100 HU to 70 HU.

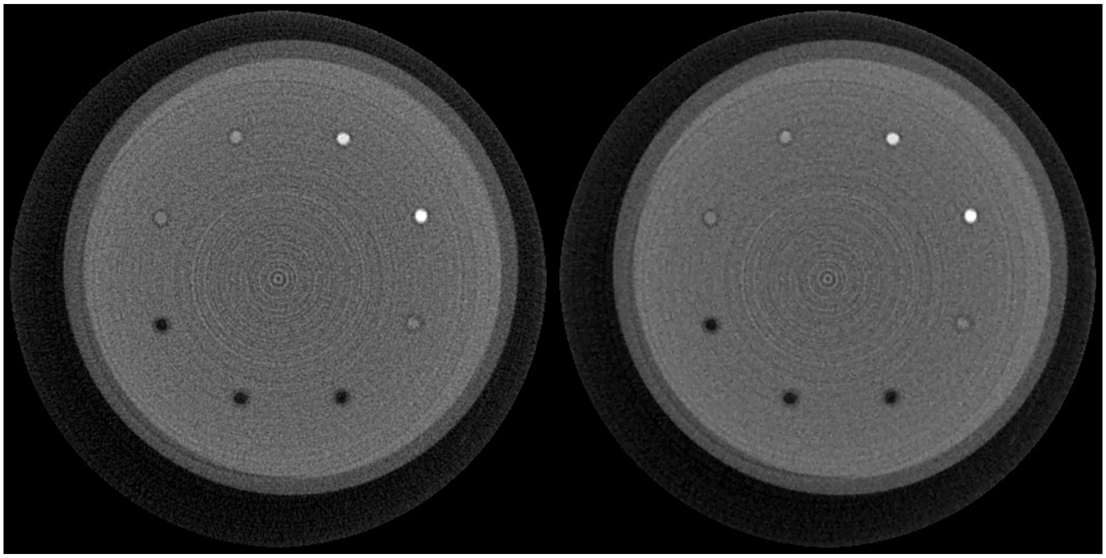

Figure 7. Slices through FBP reconstruction (left) and iterative reconstruction (right)

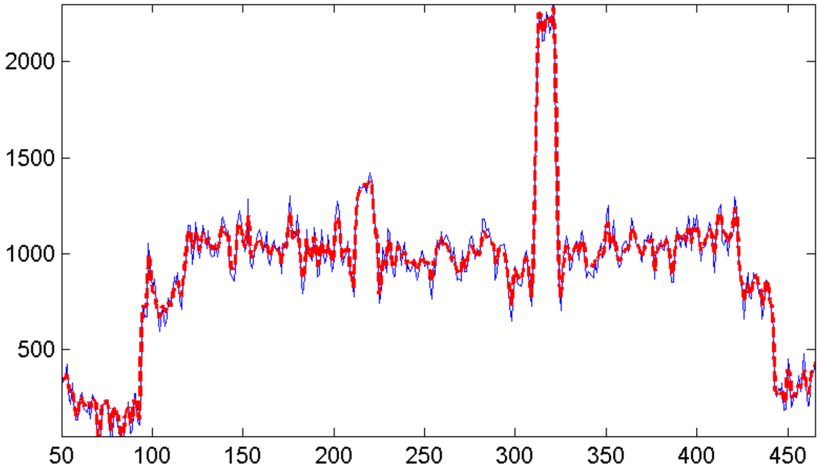

Figure 8. Cuts through reconstruction of FBP (thin, solid, blue) and iterative algorithm (thick, dashed, red).

6. CONCLUSION

In this paper, we examined the use of fast hierarchical BP/RP operators to accelerate an iterative algorithm based on conjugate-gradient minimization. We demonstrated that an iterative algorithm based on fast hierarchical algorithms can provide image quality indistinguishable from an iterative algorithm using conventional BP/RP operators, while providing almost a 10x speedup in reconstruction rates. We are currently investigating use of preconditioned CG to improve the convergence rate of the algorithm. An accelerated algorithm with low iteration counts would be an attractive option for routine use of iterative reconstruction.

ACKNOWLEDGEMENTS

The project described was supported by a Phase II SBIR Award Number R44EB005576 from the National Institute of Biomedical Imaging and Bioengineering to InstaRecon, Inc. The content of this publication is solely the responsibility of the authors and does not necessarily represent the official views of the National Institute of Biomedical Imaging and Bioengineering or the National Institutes of Health.

REFERENCES

[1] Xiao, S., Bresler, Y., and Munson, D. “Fast Feldkamp Algorithm for Cone-Beam Tomographic Reconstruction,” Proc. ICIP 2003, 819-822 (2003).

[2] Whiting B., Massoumzadeh P., et al, “Properties of preprocessed sinogram data in x-ray computed tomography”, Medical Physics 33(9), 3290-3303 (2006).

[3] De Man B., Fessler J., et al. “A study of four minimization approaches for iterative reconstruction in X-ray CT”, 2005 IEEE Nuclear Science Symposium Conference Record, 2708-2710 (2005).

[4] Sunnegardh J., Danielsson P.E., “A New Anti-Aliased Projection Operator for Iterative CT Reconstruction,” 9th International Meeting on Fully Three-Dimensional Image Reconstruction, 125-127 (2009).

[5] Brokish, J., and Bresler, Y., “Sampling Requirements for Circular Cone Beam Tomography,” 2006 IEEE Nuclear Science Symposium Conference Record, 2882-2884 (2007).

[6] www.imp.uni-erlangen.de/phantoms/