Implementation of the Fast Approximate 3D Image Reconstruction Algorithm

Tung-Kuang Wu

Department of Information Management
Minghsin Institute of Technology
HsinFong, HsinChu, Taiwan, R.O.C.
Email: tkwu@mis.mhit.edu.tw

Abstract

For an N×N×N image I(x,y,z), it would generally require $O(N^5)$ time to compute the projections of N^3 different planes, since each plane contains about N^2 sample points from I(x,y,z). Reconstruction of an N×N×N image from N^3 projections also would usually take $O(N^5)$ time. Our fast approximate algorithmth performs planar projection and backprojection in only $O(N^3 \log N)$ time. In this paper, we present the implementation results (include both timing results and images reconstructed by the algorithms) of the fast approximate projection algorithm. The results demonstrate that our algorithms are not only feasible, but also performs quite well.

Key Words: Radon Transform, 3D Image Reconstruction, Approximate Algorithms, Medical Imaging

1. Introduction

The problem of digitally (2D) reconstructing an image from its projections has become important during the past two decades. The theoretical basis of X-ray computed tomography is the mathematical theorem of Radon [9] which states that a function f(x, y) can be uniquely reconstructed from its line integrals in the x-y plane. There are many areas where practical applications of this problem arise. In the field of diagnostic radiology, this includes computed tomography, in which X-rays are used to generate the projection data for a cross section of the human body. From the projection data, one reconstructs a cross-sectional image depicting with very high resolution the morphological details of the body in that cross section. Other medical applications are Positron Emission Tomography (PET) and Magnetic Resonance Imaging (MRI).

The reconstruction of 3D volume data by 2D sectional imaging (a slice at a time) is commonly used in X-ray computed tomography. Consecutive 2D sections are stacked to form a 3D image, with the data for each section being acquired and reconstructed independently of any other section ([6], [7]). But this approach makes poor use of the available imaging photons in the case of nuclear medicine (in particular, Positron Emission Tomography) by rejecting the direction of the photons outside a single section. One of the reasons for this waste of activity was that no practical computer algorithm had been developed for reconstructing 3D images from all the data which could be acquired if the interslice collimators were omitted [11]. But the demand to increase the sensitivity of reconstructed images by making better use of the oblique rays is driving the development of 3D reconstruction of digital images.

In the case of PET, to acquire a full set of planar projections is not possible due to the geometry of the detectors. To solve the problem, some researchers ([11], [5], [4], and [13]) propose a two-pass 3D reconstruction procedure by first estimating (reconstructing) the voxel value in each grid point using the conventional 2D backprojection method, followed by a planar forward projection process to approximately compute the planar projections that are not collected by the detectors. After the full set of projections are available, a 3D reconstruction is applied. The complete procedure is as follows.

Algortihm 1. 3D Image Reconstruction Procedure

Step 1: Perform a 2D image reconstruction and obtain an estimation of the 3D voxel data set.
Step 2: Perform 2D forward projection to the 3D data to obtain a set of planar projections that are not collected by the detector.
Step 3: Perform a 3D image reconstruction using both the detector-collected projections and forward-projected planar projections (computed in Step 2).

However, reconstruction from planar projections requires that the data be *filtered* to obtain its second derivative, and *backprojected*, by summing all of the planes that "contain" a point (x,y,z) to reconstruct that point [12]. For an $N{\times}N{\times}N$ image $I(x,y,z)$, it would generally require $O(N^5)$ time to compute the projections of N^3 different planes, since each plane contains about N^2 sample points from $I(x,y,z)$. Reconstruction of an $N{\times}N{\times}N$ image from N^3 projections also would usually take $O(N^5)$ time. Studies shown in [8], [14] and [5] indicate that it takes hours to compute the above 3D image reconstruction and between 78 to 86% of the total computation time is devoted to the forward projection and backprojection phases. As a result, a fast forward/backward projection algorithm has important implications to 3D imaging, which can potentially speed up the reconstruction process.

We proposed a fast approximate algorithm for the computation of planar projections of a 3D image and the inverse problem of reconstructing a 3D image from its projections [15]. Using a new fast *Approximate Discrete Radon Transform (ADRT)* algorithm designed for the two-dimensional Radon Transform [2], we construct an algorithm (3-Dimensional *Approximate Discrete Radon Transform : 3D ADRT*) to perform approximate planar projection and backprojection in only $O(N^3 \log N)$ time. The algorithm is *approximate*, in that some of the sample points used to compute a projection may be greater than one half unit from the desired plane. However, this error is small, and we will show that the quality of reconstructied images using this technique is good. In addition, our implementation results demonstrate that the algorithms are computationally efficient, and is feasible in practice.

The remainder of this paper is organized as follows. A brief description of the 3D ADRT algorithm is given in Section 2. In Section 3 we show the image quality of our approximate 3D reconstruction algorithm. Timing results of the 3D ADRT implemenation will be shown in Section 4. Finally, we give a summary of our work and further research direction in Section 5.

2. Approximate Discrete Radon Transform & Approximate Discrete Planar Projection Algorithm

In the 2D Discrete Radon Transform, a set of summed projections is computed through a 2D image at various orientations. Consider an $N \times N$ image, $I(x,y)$. If the sampling is dense enough so that every pixel is used to compute at least one ray at any given projection angle, then the number of sequential operations needed to compute a single 2D projection will be $\Omega(N^2)$, and computing projections at N different angles (independently) will require $\Omega(N^3)$. However, for discrete non-interpolated line sampling algorithms, different orientations do not necessarily have to be computed independently. There can be a great deal of intersection between the sample points of lines at neighboring angles. For example, Figure 1 shows lines at two orientations that share half of their data points. One can potentially save time by computing such shared partial sums only once for use in two or more lines. Unfortunately, it is generally difficult to determine the proper subsets and order the computations accordingly, and it may be easier to simply calculate the sums independently. The Approximate Discrete Radon Transform ([2], [15]) defines a new line sampling algorithm that sacrifices a little accuracy and generality in order to generate line rasterizations that allow maximum sharing of intermediate terms. As a result, the ADRT is able to compute a specific set of N projections over an $N \times N$ image in only $O(N^2 \log N)$ steps (see [2], [15] for a detailed description).

Using the fast 2D Approximate Discrete Radon Transform (2D ADRT) algorithm designed for the two-dimensional Radon Transform, we construct an algorithm to compute approximate planar projections at N^2 different orientations, in only $O(N^3 \log N)$ time [15]. A total of N planes are projected at each angle, and most pass through $\Omega(N^2)$ data points. Our algorithm is thus $\Theta(N^2/\log N)$ faster than computing them independently. The planar projection algorithm can be applied to the image reconstruction problem in several ways. It can used to compute forward projection, 2D image reconstruction, and truly 3D image reconstruction. In the following, we base our discussion on Algorithm 1 (the 3D image reconstruction procedure) and show when and how the planar projection algorithm can be of use.

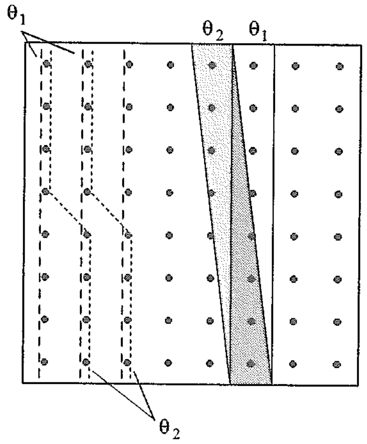

Figure 1. Overlap between neighboring lines.

Algorithm 2. Procedure for the computation of forward/backward projections and filtering.

```
Forward Projection:
Phase 1:
    for slice $I(x, y=c, z)$, $0 \le c < N$ do
    {
        for $0 \le i < 4$ do
            Compute line projections
            $P1_i(x, y=c, a)$ using the 2D ADRT
    }
Phase 2:
    for slice $P1_i(x, y, a=c)$,
    $0 \le c < N$ and $0 \le i < 4$ do
    {
        for $0 \le j < 4$ do
        Compute planar projections
        $P2_{ij}(x, b, a=c)$ using the 2D ADRT
    }
Backward Projection
Filtering:
    Compute second order difference
    $P''2_{ij}(x, b, a)$, $0 \le i, j < 4$
    from planar projections $P2_{ij}(x, b, a)$
Phase 3:
    for slice $P''2_{ij}(x, b, a=c)$,
    $0 \le c < N$, and $0 \le i, j < 4$ do
    {
        Compute projections
        $P3_{ij}(x, y, a=c)$ using the 2D ADRT
    }
    Sum partial projections
    $P3_{ij}(x, y, a=c)$, $0 \le j < 4 \Rightarrow P3_i(x, y, a=c)$
Phase 4:
    for slice $P3_i(x, y=c, a)$,
    $0 \le c < N$ and $0 \le i < 4$ do
    {
        Compute line projections
        $P4_i(x, y, z)$ using the 2D ADRT
    }
    Sum partial projections
    $P4_i(x, y, z)$, $0 \le i < 4 \Rightarrow \hat{I}(x, y, z)$
```

The ADPP algorithm can be used in the forward projection process (Step 2 of Algorithm 1). It happens that the same algorithm can also be used in the backward projection process (Step 3, see [15] for the proof). Note that if Step 1, the 2D image reconstruction, is computed using the 2D ADRT, the whole 3D image reconstruction procedure then contains repeated applications of the 2D ADRT, plus some filtering steps ([15]). This greatly simplifies the implementation of the 3D reconstruction process.

The detailed procedure for the computation of the full set of forward and backward projections (plus the filtering step) of the 3D ADRT is given in Algorithm 2. We also present the forward projection procedure, since it is needed to compute forward planar projections for use with our study. $I(x, y, z)$ and $\hat{I}(x, y, z)$ represent the original and the reconstructed 3D image respectively. Note that 20 (rather than 32) applications of the 2D ADRT (to each of N slices of size $N \times N$) are required in the forward or backward projection process (see [15] for a detailed description).

The 3D image reconstruction procedure can be performed starting from the filtering step in Algorithm 2 if a full set of planar projections are available. However, the ADPP-based algorithm can also be used to compute the 3D reconstruction if only line projections are available. By starting from the second phase of Algorithm 2, we forward-project the line projections to 2D planar projections. Once the planar projections are computed, a 3D image reconstruction can be performed. The computational complexity remains $O(N^3 \log N)$, however the complicated filtering steps required in 2D image reconstruction are avoided. Note that the data must be collected in linogram fashion (see [1] for approaches in collecting Linogram projections) or interpolated into the proper sampling distribution.

3. Quality of Reconstructed Images using the Approximate Dsicrete Planar Projection (ADPP) Algorithm

We have evaluated the 3D reconstruction algorithm (Algorithm 2) with an artificially constructed 64×64×64 3D image, containing 7 spheres. The planar projections (forward projections) are computed using the ADPP algorithm. The filter is the simple second order difference operator described in previous section. Figure 2 shows a sequence of slices of the original and reconstructed 3D images (each of size 64×64). The quality of the reconstructed images appears to be fairly good. Figure 3 shows the pixel intensities vs. position along lines of slice 31. The reconstructed image follows closely the original data, except at very high frequency points. (This is typical of reconstructions by standard techniques as well.)

4. Performance of the 3D ADRT Implementation

The Approximate Discrete Planar Projection algorithm has been implemented and evaluated on various sequential machines. Our test image is an artificially constructed

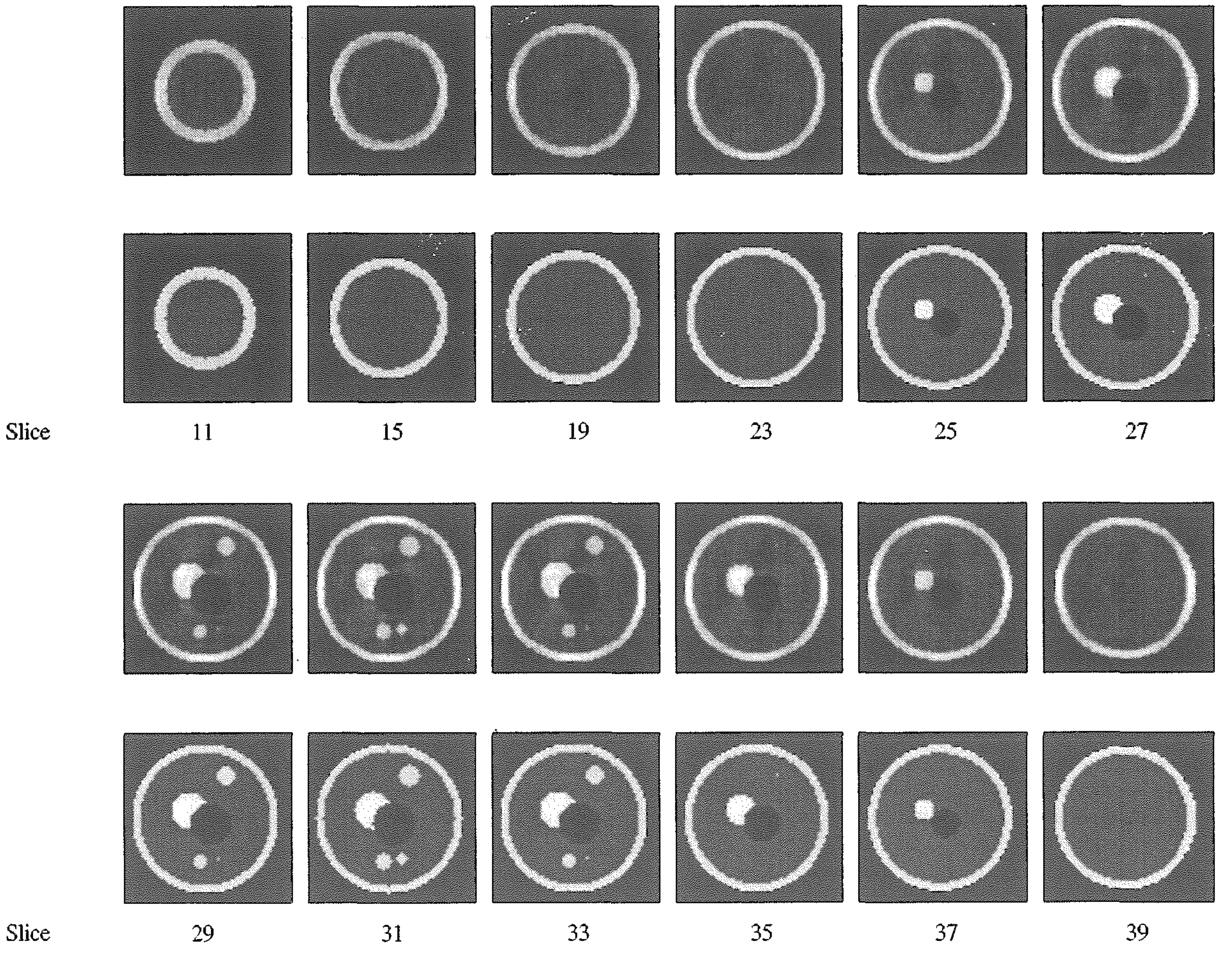

Figure 2. A sequence of slice images (each with size 64×64) from the original and reconstructed 64×64×64 phantom (containing 7 spheres) (top: reconstructed images, bottom: original images)

64×64×64 "phantom" (as described in Section 3), interpolated into 16×16×16, 32×32×32, 64×64×64, and 128×128×128 voxel data sets. Forward projections of the phantom are computed using our approximate planar projection algorithm. We then reconstruct the 3D image by computing the filtered backprojection, with the same approximation algorithm used in the forward projection process. A second order difference is used as the filter.

The procedure of our implementation of 3D reconstruction was given in Section 2 (Algorithm 2). In planar 3D image reconstruction, the reconstruction process starts from the filtering step since the planar projections are already available. Otherwise, the reconstruction is started from phase 2 if the data are collected in a line projection format.

Note that zero-padding is used in the implementation of the ADRT algorithm [15]. Although the ADRT algorithm is applied to slices of 3D voxel data in each phase of the algorithm, we expand (by zero-padding) the N×N×N data into a 2N×2N×2N voxel data set (instead of expanding slices of N×N data to 2N×N). The data size is therefore increased by a factor of 8, rather than 2. The reason for adopting such a strategy is for convenience in later steps which need to sum partial projections of different ranges to compute complete projections. This is, however, mainly an implementation issue. A smarter implementation might reduce the padding and improve the performance by about 4 times. We present results of our implementations of the approximate 3D image reconstruction below.

The platform we used is a IBM Compatible PC with Pentium III 450 MHz CPU and 256 MB main memory, running Windows NT Server 4.0. We also perform 3D image reconstruction from 1D line projections on a DEC 5000 workstation (running Ultrix 4.2 and 64 MB main memory) for a comparison to the performance data derived in previous research [10] which implemented 2D image reconstruction based on the 2D ADRT algorithm.

Table 1. Execution time (in real time, ignoring I/O) of 3D ADRT image reconstruction on a Pentium III 450 PC. (All times given in seconds.)

Data Size	Backprojection	Filtering
32×32×32	1.08	0.15
64×64×64	10.8	1.2
128×128×128	146.2	14.9

Table 1 shows the time required (real time in seconds, on the Pentium III PC) by our algorithm for 3D image reconstruction, separated into backprojection and filtering times. The backprojection execution time includes phase 3 and 4 of Algorithm 2. In each case, the input data contains 16N–16 projections, with 2N–2 radial and 2N–2 angular angles per projection. Since there is no comparable algorithm, we show some earlier published timing results for 3D backprojection using other algorithms and platforms in Table 2 for a rough comparison. The many differences in hardware and data sizes make direct comparison impossible, however it seems reasonable to assume that our method will be extremely fast in practice compared to standard techniques.

Recall that the ADPP algorithm can be used to perform 2D image reconstruction by initiating the reconstruction procedure from phase 2 of Algorithm 2, if there are only 1D linear projections available. To demonstrate the feasibility of the idea, we make a performance comparison between the two approaches (ours, and the work in [10] which uses 2D ADRT and standard reconstruction approach), as shown in Table 3. For $N \leq 32$ our approach exhibits close to two times speedup by avoiding the 2D filtering step. However, notice that the execution time is larger than expected when N=64. Since we do not find a similar pattern on the Pentium III 450 PC and SUN 10 implementation (Table 4),· we suspect that the pheonomon is platform dependent. We have tested the same code in two other platforms — a SUN 4 and a SUN 10 workstation — each with the same main memory (64 Mbyte) as the DEC 5000 workstation. The results are shown in Table 4. Interestingly, the results on SUN 4 do show the same pattern. On the other hand, the results on SUN 10 follow the expected asymptotic performance increase.

Table 2. Published computation times for 3D backprojection;. (All times given in seconds.) (* # of sinograms/# of radial angles/# of angular angles; ** # of sinograms/# of combined radial and angular angles; † With Vaccelerator board; ‡ Time for whole reconstruction procedure, i.e., time for both filtering and backprojection.)

References	Platform	Input data size	Image generated	Exec. time
3DADRT	SUN 10	N/A	64×64 ×64	304.4
[5]	SUN 4	256/128 /96*	128×128 ×31	13800
[4]	VAX 3200†	58/128 /160*	128×128 ×15	1380
[13]	SUN SPARC 2	72/3372 **	320000 voxels	3240
[3]	44-node T-800 Transputer	256/128 /96*	128×128 ×31	1174‡

Table 3. Execution time (in CPU seconds, ignoring I/O) of ADRT-based 3D and 2D image reconstruction using both the ADRT and standard (STD) algorithms on a DEC 5000 workstation. (* times derived by multiplying the reconstruction time of a N×N image by N)

Data Size	Ours – 3DADRT	[10] - ADRT	[10] - STD
16×16×16	4.7	11.2*	16.0*
32×32×32	44.9	83.2*	147.2*
64×64×64	607.8	697.6*	985.6*

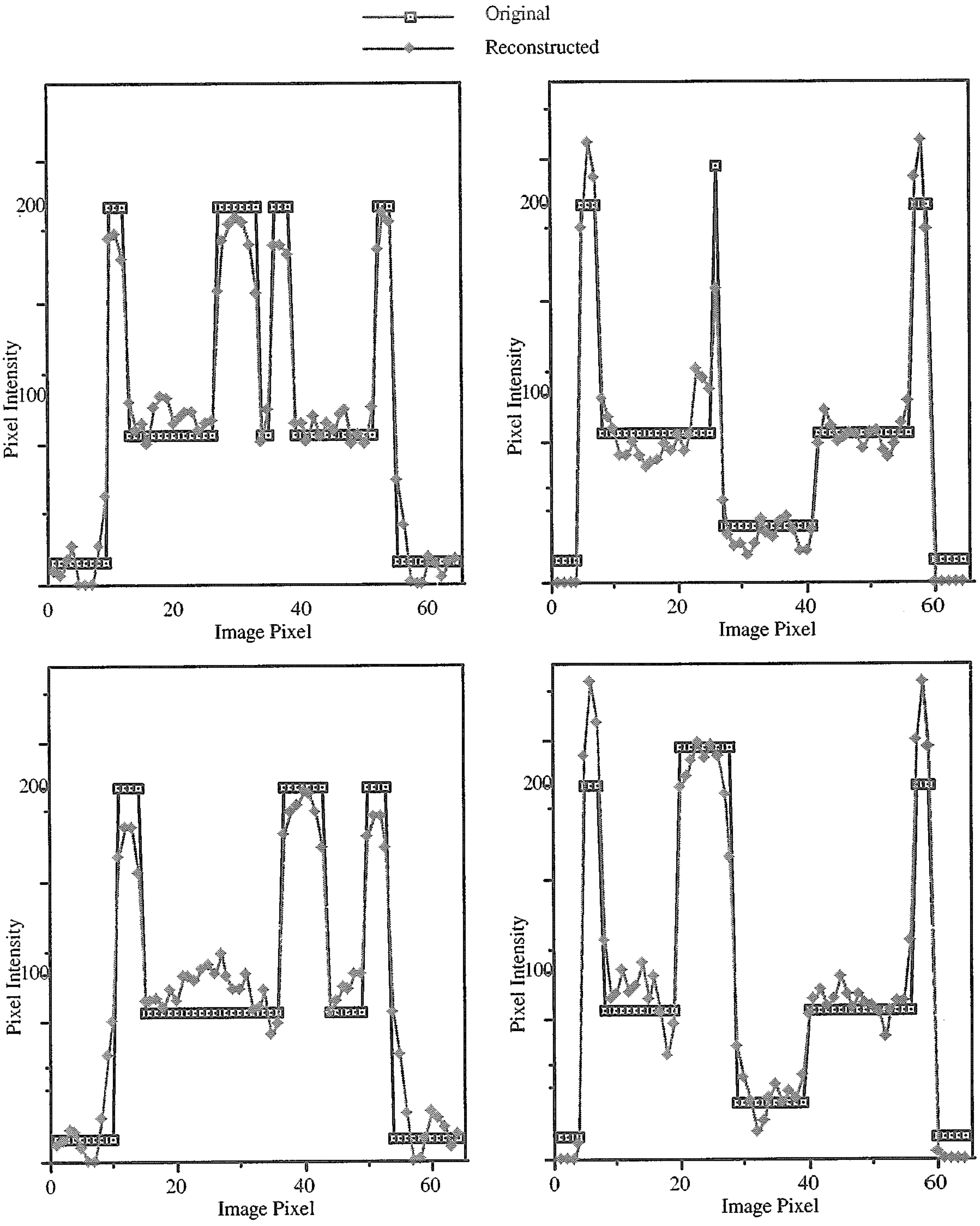

Figure 3. Plots of image pixels vs. pixel intensity for row 15 (top left) and row 26 (top right), row 34 (bottom left) and row 48 (bottom right) of slice 31.

Table 4. Execution time (in CPU seconds, ignoring I/O) of ADRT-based 3D algorithm on SUN 4 and SUN 10 workstations.

Data Size	SUN 4	SUN 10	P-III PC
16×16×16	6.5	3.5	0.17
32×32×32	60.1	32.6	1.83
64×64×64	776.0	304.4	18.29

5. Summary

In this paper we have presented an asymptotically fast approximate planar forward/backward projection algorithm which, when combined with a simple second order difference filter, can compute truly 3D image reconstruction (from projections) very efficiently. This algorithm is directly applicable to medical imaging techniques that collect planar projection data directly, such as in Magnetic Resonance Imaging (MRI) and Positron Emission Tomography (PET). In this case, the data must either be collected in Linogram fashion [1] or interpolated into the proper sampling distribution. The fast filtered backprojection can then be applied. However, the approach can also be applied to data collection techniques that produce line projection data. Often, such data are collected as independent 2D slices, and reconstructed independently using 2D reconstruction methods. Normally, this would require $O(N^4)$ time to reconstruct data with N angles, N projections per angle, and N slices. Furthermore, it requires a more complex (and in practice, time-consuming) filtering operation, a convolution with the function whose Fourier transform is $|\omega|$. Note that after the first phase of our *forward* projection algorithm, we obtain this same type of data. We can therefore apply our method to this data by starting from the second phase of the forward projection algorithm to compute planar projections from the line projections. We then compute a full 2-pass 3D reconstruction as before. The backprojection time is $O(N^3 \log N)$. Furthermore, the relatively complicated Fourier domaing filtering is replaced by a simple local second order difference filter, requiring only $O(N^3)$ time [12].

The effects of using different filtering techniques on the quality of the reconstructed images deserve further evaluation. A qualitative comparison of the reconstructed images between the ADPP-based 3D image reconstruction and ADRT-based reconstruction methods would also be interesting. Finally, there might exist other approximate line sampling techniques, which may be potentially faster and more efficient than the ADRT-based one. An approximate curvature (non-planar) sampling technique should have many potential applications in the fields of image processing and computer vision.

6. Reference

[1] L. Axel, G. T. Herman, and D. Roberts, "Linogram Reconstruction for Magnetic Resonance Imaging (MRI)," *IEEE Trans. on Medical Imaging*, 9 (4), pp. 447-449 (1990).

[2] M. L. Brady, "A Fast Discrete Approximation Algorithm for the Radon Transform," SIAM Journal on Computing, 27(1), pp. 107-119, (Feb. 1996).

[3] C. Comtat, C. Morel, M. Defrise, and D. W. Townsend, "The FAVOR Algorithm for 3D PET Data and Its Implementation Using a Network of Transputers," *Physics in Medicine and Biology*, Vol. 38, pp. 929-944 (1993).

[4] S. R. Cherry, M. Dahlbom, and E. J. Hoffman, "Evaluation of a 3D Reconstruction Algorithm for Multi-slice PET Scanners," *Physics in Medicine and Biology*, 37 (3), pp. 779-790 (1992).

[5] M. Defrise, D. Townsend, and A. Geissbuhler, "Implementation of Three-Dimensional Image Reconstruction for Multi-ring Positron Tomographs," *Physics in Medicine and Biology*, Vol. 35, pp. 1361-1372 (1990).

[6] G. T. Herman, Image Reconstruction from Projections, (New York: Academic Press), 1980.

[7] A. C. Kak, "Image Reconstruction from Projections," *Digital Image Processing Techniques*, Academic Press (1984).

[8] P. E. Kinahan, and J. G. Rogers, "Analytic 3D Image Reconstruction Using All Detected Events," *IEEE Trans. on Nuclear Science*, 36 (1), pp. 964-968 (1989).

[9] J. Radon, "Über die Bestimmung von Funktionen durch ihre integralwerte längs gewisser Mannigfaltigkeiten," *Ber. Saechs. Akad. Wiss. Leipzig, Math-Phys. Kl.*, 69, pp. 262-277 (Apr. 1917).

[10] D. A. Ramakrishnan, Master Thesis, Department of Electrical and Computer Engineering, Penn State University, (Jan. 1993)

[11] J. G. Rogers, R. Harrop, and P. E. Kinahan, "The Theory of Three-Dimensional Image Reconstruction for PET," *IEEE Trans. on Medical Imaging*, Vol. MI-6, pp. 239-243 (1987).

[12] L. A. Shepp, "Computerized Tomography and Nuclear Magnetic Resonance," *Journal of Computer Assisted Tomography*, 4 (1), pp. 94-107 (1980).

[13] M. W. Stazyk, J. G. Rogers, and R. Harrop, "Full Data Utilization in PVI Using the 3D Radon Transform," *Physics in Medicine and Biology*, Vol. 37 (3), pp. 689-704 (1992).

[14] D. W. Townsend, T. Spinks, T. Jones, A. Geissbuhler, and M. Defrise, "Three Dimensional Reconstruction of PET Data from a Multi-ring Camera," *IEEE trans. on Nuclear Science*, Vol. 36 (1), pp. 1056-1065 (1989).

[15] T. K. Wu, and M. L. Brady, "A Fast Approximation Algorithm for 3D Image Reconstruction", *Proceedings of the 1998 International Computer Symposium (Workshop on Image Processing)*, (Dec., 1998).