Fast Feldkamp Reconstruction Based on Focus of Attention and Distributed Computing

J. Gregor,[1] S. S. Gleason,[2] M. J. Paulus,[2] J. Cates[3]

[1] Department of Computer Science, University of Tennessee, 1122 Volunteer Boulevard, Suite 203, Knoxville, Tennessee 37996-3450; E-mail: jgregor@cs.utk.edu

[2] ImTek, Inc., 3200 Mynatt Avenue, Knoxville, Tennessee 37919; E-mail: gleasons@imtekinc.com, paulusmj@imtekinc.com

[3] School of Computing, University of Utah, 50 S. Central Campus Drive, Salt Lake City, Utah 84112-9205; E-mail: cates@sci.utah.edu

Received 14 June 2002; accepted 18 November 2002

ABSTRACT: The Feldkamp algorithm is widely accepted as a practical conebeam reconstruction method for three-dimensional x-ray computed tomography. We introduce focus of attention, an effective and simple to implement datadriven preprocessing scheme, for identifying a convex subset of voxels that include all those relevant to the object under study. By concentrating on this subset of voxels during reconstruction, we reduce the computational demands of the Feldkamp algorithm correspondingly. To achieve further speed-up, all computations are distributed across a cluster of inexpensive, dual-processor PCs. We present experimental work based on mouse data obtained from the MicroCAT which is a high-resolution x-ray computed tomography system for small animal imaging. This work shows that focus of attention can cut the overall computation time in half without affecting the image quality. The method is general by nature and can easily be adapted to apply to other geometries and modalities as well as to iterative reconstruction algorithms. © 2003 Wiley Periodicals, Inc. Int J Imaging Syst Technol, 12, 229–234, 2002; Published online in Wiley InterScience (www.interscience.wiley.com). DOI 10.1002/ima.10027

Key words: x-ray computed tomography; conebeam imaging; filtered backprojection; support function estimation; distributed computing

I. INTRODUCTION

Three-dimensional x-ray computed tomography (CT) is becoming essential in many applications including nondestructive testing and reverse engineering of industrial parts for quality control and assurance, characterization and evaluation of simple and composite materials, and non-invasive imaging in the biomedical sciences for the purpose of screening small animals for pathologies and mutations as well as monitoring disease progression and response to therapy. Although exact conebeam reconstruction algorithms exist, a so-called approximate algorithm is often preferred in practice (Wang et al., 1999). When the scanning locus is circular, the filtered backprojection based Feldkamp algorithm (Feldkamp et al., 1984) is widely accepted as a practical conebeam reconstruction method. Computationally, the voxel-driven backprojection constitutes the most expensive step of the algorithm. A divide-and-conquer scheme for reducing the time complexity of the backprojection computation has been proposed (Turbell and Danielsson, 1999), but the speed-up comes at the cost of an increase in the memory requirements. Fortunately, the standard Feldkamp algorithm can be parallelized (Reimann et al., 1996; Laurent et al., 1996) making it possible to achieve reasonable reconstruction times by distributing the computation across a cluster of PCs.

The conebeam geometry is such that only the voxels located inside the axial cylinder inscribed by the image volume need be considered (Kak and Slaney, 2001). In this article, we show that it is possible to identify an even smaller, but still convex subset of voxels that include all those relevant to the object under study. We call the approach focus of attention, and the general idea behind this data-driven preprocesing scheme is as follows. Consider shining a light at an object from a number of different view angles. When backprojecting the resulting shadowgrams into the image space, the intersection thereof ideally forms a convex hull. Only the projection and image data that pertain to this hull need be considered for the reconstruction, reducing the amount of computation needed. Distributed computing is accommodated in a straightforward manner. We have previously presented focus of attention in connection with iterative, parallel-beam algorithms for 2D positron emission tomography (Gregor and Huff, 1997, 1998) and 3D projection magnetic resonance imaging (Gregor and Rannou, 2001, 2002). This work was based directly on results from the theory of support functions (Prince and Willsky, 1990; Karl et al., 1995). Because of the

Correspondence to: J. Gregor, Department of Computer Science, University of Tennessee, 1122 Volunteer Boulevard, Suite 203, Knoxville, TN 37996-3450; E-mail: jgregor@cs.utk.edu

Contract grant sponsor: National Institutes of Health; *Contract grant number:* 1 R01 EB00789-01A2

© 2003 Wiley Periodicals, Inc.

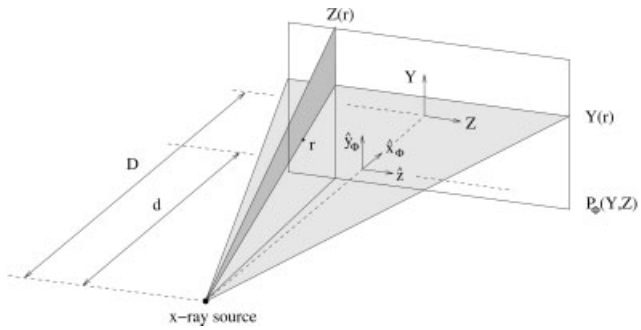

Figure 1. Conebeam geometry. The x-ray source and the detector array rotate around the $\hat{z}$ axis of the image volume. The projection of voxel **r** onto the detector array is given by the intersection of the orthogonal fans shown in light gray.

divergent nature of the conebeam geometry, we take a slightly different approach here. This new approach, which is conceptually simple but of significant practical value, is less noise sensitive and much easier to compute than its older counterparts. Furthermore, although developed for conebeam x-ray CT, it is general by nature and can easily be adapted to apply to other geometries and modalities as well as to iterative reconstruction algorithms.

We present experimental work based on mouse data obtained from the high-resolution MicroCAT™ (ImTek, Inc.) small animal imaging system. Originally developed at Oak Ridge National Laboratory (ORNL) in support of the Mammalian Genetics Research Facility (affectionately known as the "Mouse House") (Gleason et al., 1999; Paulus et al., 1999, 2000), the MicroCAT has since been transferred to industry for commercialization. The proposed focus of attention preprocessing scheme together with the parallelization of it and the Feldkamp algorithm has been developed to support research in which high-throughput micro-CT scanning and reconstruction is essential.

II. FELDKAMP ALGORITHM

Let f denote a 3D image volume to be reconstructed, and let $\mathbf{r}^T = (x, y, z)$ be the spatial coordinates of a particular voxel. Furthermore, let $P_\Phi(Y, Z)$ represent a 2D projection with (Y, Z) denoting the spatial location of a particular detector element. Note that $P_\Phi = -\log(I_\Phi - I_D)/(I_0 - I_D)$, where I_Φ is the photon flux recorded for view angle Φ with an animal present, and I_0 and I_D are blank and dark current projections used for normalization purposes. Let $\hat{x}_\Phi$, $\hat{y}_\Phi$, and $\hat{z}$ denote the image axes rotated to line up with the given projection. Finally, let d denote the distance from the x-ray source to the axis of rotation, i.e., the $\hat{z}$ axis of the image volume, and let D be the distance from the x-ray source to the center of the detector array. See Figure 1 for an illustration of the variables and their geometric relations.

Following standard practice, we can make the simplifying assumption that $D = d$, since scaling by d/D converts data from the actual geometry to this idealized set-up. This allows us to express the Feldkamp algorithm (Feldkamp et al., 1984) in terms of the following weighted, filtered backprojection computation:

$$f(\mathbf{r}) = \frac{1}{4\pi^2}\int_0^{2\pi} \frac{d^2}{(d + \mathbf{r}\cdot\hat{x}_\Phi)^2}\,\hat{P}_\Phi\,(Y(\mathbf{r}), Z(\mathbf{r}))\,d\Phi,$$

where

$$Y(\mathbf{r}) = \frac{\mathbf{r}\cdot\hat{y}_\Phi}{d + \mathbf{r}\cdot\hat{x}_\Phi}\,d, \quad Z(\mathbf{r}) = \frac{\mathbf{r}\cdot\hat{z}}{d + \mathbf{r}\cdot\hat{x}_\Phi}\,d,$$

$$\hat{P}_\Phi\,(Y, Z) = \frac{d}{\sqrt{d^2 + Y^2 + Z^2}}\,P_\Phi(Y, Z) ** g(Y),$$

and g represents a ramp-like filter kernel. For the results in this article, we approximate the backprojection and convolution integrals by Riemann sums, index the filtered projection data via bilinear interpolation, and use a Shepp-Logan convolution filter kernel.

III. FOCUS OF ATTENTION

The theory of support functions (Prince and Willsky, 1990; Karl et al., 1995) provides criteria for establishing a consistent mapping between the projection space and the image space. Our earlier work on focus of attention for parallel-beam geometries was based explicitly on these criteria (Gregor and Huff, 1997, 1998; Gregor and Rannou, 2001, 2002). With reference to Figure 2, consider a 2D object $\mathbb{O}$ and a unit direction vector $\mathbf{u}_\theta^T = (\cos\theta, \sin\theta)$ anchored somewhere therein. The concept of a support value can then be defined as the maximum distance from the anchor point to any point contained within the object measured in the direction of the unit vector. Mathematically,

$$h(\mathbf{u}_\theta) = \max_{\mathbf{r}\in\mathbb{O}} \mathbf{r}\cdot\mathbf{u}_\theta.$$

Extention to 3D is straightforward. Now let vector **h** denote a collection of support values, and let matrix **Q** represent determinantal conditions with respect to the support values obtained for neighboring unit direction vectors. Convexity of $\mathbb{O}$ then implies that the vector inequality $\mathbf{Qh} \le 0$ must hold. As an example, consider the case of 2D parallel beam geometry for which we would use the $M \times M$ Toeplitz matrix (Prince and Willsky, 1990):

$$\mathbf{Q} = \begin{bmatrix} 1 & -k & 0 & \cdots & 0 & -k \\ -k & 1 & -k & 0 & \cdots & 0 \\ 0 & -k & 1 & & & \vdots \\ \vdots & 0 & & \ddots & & 0 \\ 0 & \vdots & & & 1 & -k \\ -k & 0 & \cdots & 0 & -k & 1 \end{bmatrix} \tag{1}$$

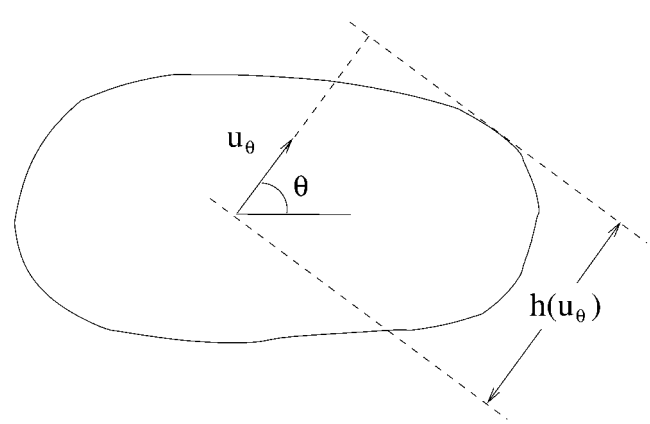

Figure 2. Support value $h(\mathbf{u}_\theta)$ is a distance measured in the direction of unit direction vector $\mathbf{u}_\theta$.

where M is the number of equally spaced unit direction vectors and $k = 1/(2 \cos 2\pi/M)$. For large M, the implication is that convexity is guaranteed if all support values are less than or equal to the average of their neighbors. The extension to a 3D parallel beam geometry is quite complicated. Suffice it here to say that matrix $\mathbf{Q}$ also in this case captures the local geometric context of each support value. See the paper by Karl and coworkers (1995) for mathematical details. Alternatively, the reader may refer to a computationally oriented treatment of the subject matter given in a recent article by Gregor and Rannou (2002).

Our previous focus of attention approach for 2D and 3D parallel beam geometries was to compute a (noisy) support vector estimate, say $\mathbf{h}^0$, by means of moving average based thresholding of the projection data, and then find the closest (in a Euclidian sense) true support vector, say $\mathbf{h}^*$, by solving the quadratic programming problem $\min 0.5\|\mathbf{h} - \mathbf{h}^0\|^2$, with the solution appropriately constrained by the above vector inequality. Having computed $\mathbf{h}^*$, we would determine the corresponding convex hull in the image space, with the end result being a consistent segmentation that supports the elimination of equations and unknowns from the linear system of equations that form the basis for most iterative reconstruction algorithms.

The conebeam geometry is divergent in two dimensions. This makes the above constrained optimization approach difficult to apply. However, as we now proceed to show, it is possible to achieve the desired focus of attention result without explicitly expressing and using a constraint matrix. In order to do that we relax the goal of computing the tightest possible 3D convex hull that envelops the object. We aim instead to determine the tightest fitting axial cylinder that has a convex cross-section. We furthermore solve the associated 2D support function estimation problem by means of a *shape-based* method. The 3D convex hull ultimately used for focus of attention is thus obtained by stacking identical copies of a 2D convex hull that we compute as follows.

We first apply moving average based thresholding to determine the minimum and maximum values of Y to be considered for each 2D projection. That is,

$$Y_\Phi^- = \text{argmin}_Y \sum_w P_\Phi(Y, \cdot) \geq \varepsilon_1$$

$$Y_\Phi^+ = \text{argmax}_Y \sum_w P_\Phi(Y, \cdot) \geq \varepsilon_1$$

where the summations indicate that we threshold on basis of the activity within a sliding window of width w and ε_1 is a user-defined threshold parameter. Note that the computation determines minimum and maximum values of Y without regard to Z. See Figure 3 for an illustration. Now let f_B denote a cross section of the image volume that results from backprojecting the corresponding binary versions of the projections. Specifically,

$$f_B(\mathbf{r}_{xy}) = \frac{1}{2\pi} \int_0^{2\pi} B_\Phi(Y(\mathbf{r}_{xy}))\, d\Phi,$$

where

$$B_\Phi(Y) = \begin{cases} 1 & \text{if } Y_\Phi^- \leq Y \leq Y_\Phi^+ \\ 0 & \text{otherwise} \end{cases},$$

and $\mathbf{r}_{xy}$ is the (x, y)-subvector of $\mathbf{r}$. Furthermore, let $f_{\Omega'}$ be a thresholded version of f_B with Ω' denoting the set of non-zero pixels. That is,

$$f_{\Omega'}(\mathbf{r}_{xy}) = \begin{cases} 1 & \text{if } f_B(\mathbf{r}_{xy}) \geq \varepsilon_2 \\ 0 & \text{otherwise} \end{cases}$$

where ε_2 is a user-defined threshold parameter. We then compute the support vector for $f_{\Omega'}$ and use it to determine the (small number of) pixels that potentially must be added to Ω' in order to obtain Ω, the desired convex hull. That is,

$$\Omega = \{\mathbf{r}_{xy} : \mathbf{r}_{xy} \cdot \mathbf{u}_\theta \leq h_{\Omega'}(\mathbf{u}_\theta) \text{ for } 0 \leq \theta \leq 2\pi\}$$

where

$$h_{\Omega'}(\mathbf{u}_\theta) = \max_{\boldsymbol{\sigma} \in \Omega'} \boldsymbol{\sigma}_{xy} \cdot \mathbf{u}_\theta.$$

See Figure 4 for an illustration of cross-section image f_B and associated raw and convex Ω point sets.

This concludes the preprocessing part of the focus of attention scheme. During image reconstruction, the backprojection is computed only for the convex 3D subset of voxels $\{\mathbf{r} : \mathbf{r}_{xy} \in \Omega\}$. Since we do not alter the original projection data in any way, the corresponding values of the reconstructed image, $f(\mathbf{r})$, are identically the same for the Feldkamp algorithm whether we apply focus of atten-

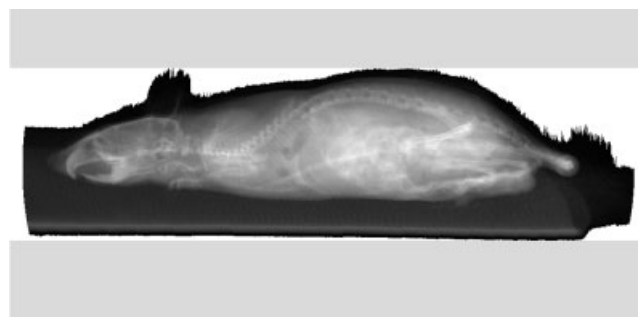

Figure 3. Moving average based thresholding of the individual projection data columns produces a ragged seg-mentation as illustrated by the black outline in the white region. The proposed focus of attention method uses the extreme values indicated by the light gray regions.

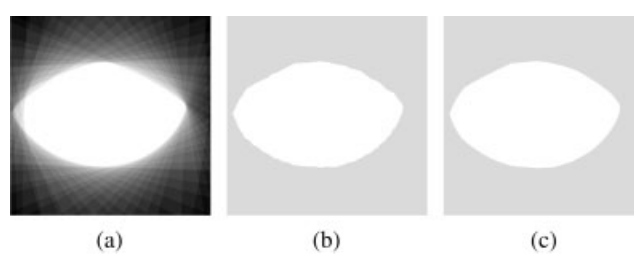

Figure 4. Cross section of reconstruction hull. (a) Example of f_B, the image that results from backprojecting the thresholded projections, (b) point set Ω', which exhibits several concavities, and (c) derived convex hull Ω on basis of which background voxels are eliminated from the reconstruction.

tion or not. The integrity of interior object voxels is thus not compromised even if we were to exclude some of the peripheral object voxels. Background voxels may, of course, be assigned different values in the two cases.

This new approach to focus of attention is less noise sensitive and much easier, and thus faster, to compute than its older counterparts. Robustness and speed is achieved by not first computing a support vector *estimate* that merely initializes a constrained quadratic programming problem, but instead directly extracting a *true* support vector from an easy to compute cross-section image that allows all projection threshold boundaries to be considered simultaneously.

With respect to the two user-defined threshold parameters, then we use $\varepsilon_1 = 0.05$ and $\varepsilon_2 = 0.95$ for the results in this article. Regarding the former threshold, then one would set it to 0.0 for ideal data. But because we deal with real data, which is noisy, we use a small, nonzero value to prevent the moving average based thresholding of the projection data from terminating prematurely since that might have the effect of greatly enlarging the cross-section of the convex-hull. Regarding the latter threshold, we point out that a value of 1.0 by definition leads to $\Omega' = \Omega$. By using a slightly smaller value, we ensure that $\Omega' \subset \Omega$ with a moderate enlargement of the convex hull cross-section being the conservative result.

IV. DISTRIBUTED COMPUTING

Several cluster implementations have been reported for the Feldkamp algorithm, typically emphasizing a clever communication scheme. One approach (Reimann et al., 1996) had a master node do all the weighting and filtering of the projections before sending the data to the other nodes each of which then performed the backprojection for its assigned image subvolume. Another approach (Laurent et al., 1998) distributed the projections across all nodes, had each node weight and filter its projection data, and then used a send-receive scheme that allowed each node the opportunity to ultimately backproject all projection data for its assigned image subvolume.

We take a simpler and easier to implement approach here. Each node is made responsible for reconstructing a particular image subvolume, the axial width of which is automatically determined by the node's rank in the computation. The corners of the bounding box corresponding to the image subvolume are projected out to the detector space to determine the subset of the projection data needed locally. This data is then weighted, filtered, and backprojected without the need for communicating anything with the other nodes. A small overhead is incurred by having some projection data being weighted and filtered on several nodes but the associated cost is somewhat offset by not having the nodes exchange the rather large projection data sets. When adding the focus of attention preprocessing to the computation, a global reduction operation is used to determine and broadcast the value of Y_Φ^- and Y_Φ^+. This data exchange takes place in logarithmic time and is thus fast.

V. EXPERIMENTAL RESULTS

We have implemented the described algorithms in C using the LAM implementation of MPI to handle the distributed computing aspects. Our cluster consists of thirty inexpensive dual-processor PCs, each equipped with 500 MHz Pentium IIIs and 512 Mbytes of memory. The PCs are interconnected via 100 Mbps FastEthernet.

Our experimental work is based on mouse data. We used a MicroCAT II (ImTek, Inc., Knoxville, TN), which was configured to have a conebeam angle of 18.26°. Projections were acquired for every 1° up to a full 360° rotation of the gantry. Each 2048×4088 projection was filtered and downsampled to 512×1022 with a resulting effective detector element size of 161.8 μm. This configuration supports the reconstruction of a $512 \times 512 \times 1022$ image volume with cubic 120.9 μm voxels. Higher resolution projections and images are possible by not downsampling the data as was done here.

To illustrate the image quality as well as to provide a visual indication of the computational savings associated with the focus of attention (FOA) method, Figure 5 shows a coronal and a transaxial slice of a reconstructed mouse image. The mouse received an intraperitoneal injection of a water-soluble iodinated contrast agent prior to being scanned. The cross-section of the reconstruction hull covers 78,038 pixels or 2.63 times fewer than the 205,887 pixels covered by the indicated inscribed circle that defines the imaging

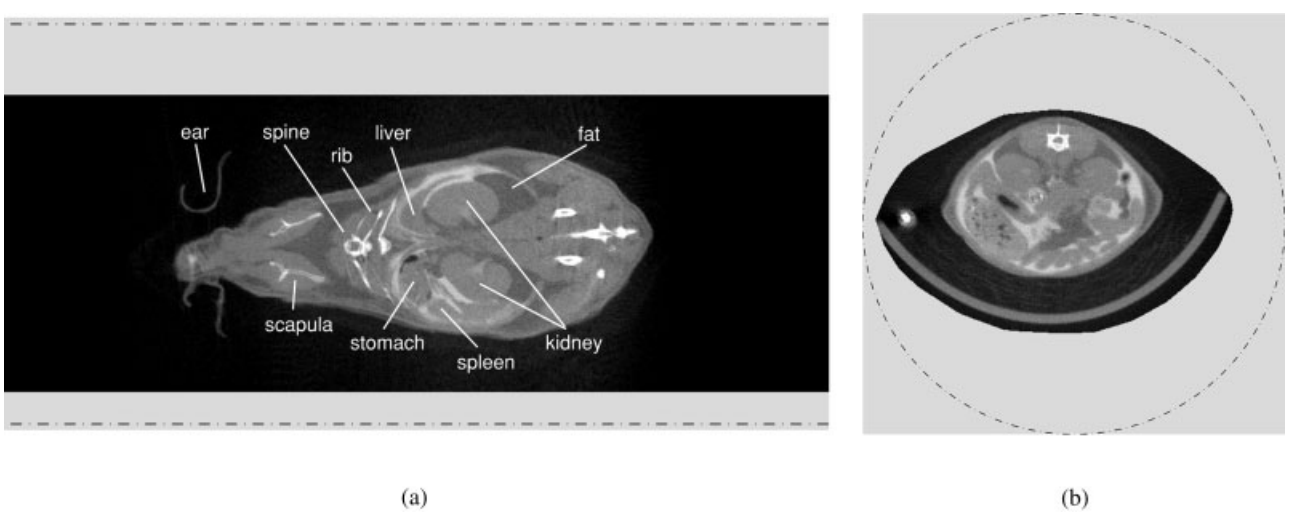

Figure 5. Reconstruction example. (a) Annotated 512×1022 coronal image and (b) 512×512 transaxial image. The light gray regions surrounding the mouse data indicate voxels not considered during reconstruction. The bright spot to the left in (b) is the tail. The dashed black lines superimposed in both images indicate the inscribed circle normally used to define a computational region-of-interest.

field of view normally considered. In the following, this inscribed circle is used to cut back on the computational cost when FOA is not being used.

Figure 6 provides a log-log plot of the wall-clock time required for a complete reconstruction as a function of the number of dual-processor PCs used. First, we notice that FOA based reconstruction times (lower plot) are about half of those without FOA (upper plot). Secondly, we see that a 7-minute reconstruction time is obtained when using FOA and all available dual-processor PCs. To place this result in context, extrapolation indicates that a single processor implementation without FOA would take on the order of 13+ hours—or 120 times longer. The achieved reduction in wall-clock time is thus quite substantial.

Tables I and II provide the average amount of time a processor spent on the tasks listed. As above, N refers to the number of dual-processor PCs used; the corresponding number of CPUs is twice that. We see that the time spent on handling the projection data, i.e., loading it from disk, normalizing, weighting and filtering it, does not vary significantly with the cluster size. We also see that the cost associated with FOA is quite small since the projection data handling takes about the same time whether this computation is included or not. With respect to the time spent on backprojection, we see that FOA results in a factor 2.48 gain in speed. This is in close agreement with the above 2.63 ratio of voxels considered.

Linear regression on the wall-clock timing data indicates that near-linear speed-up is achieved even when using a relatively large number of processors. As more and more processors are included, however, the time associated with the projection data handling begins to account for a larger and larger percentage of the overall time thereby effectively causing the speed-up to eventually taper off and become sublinear. The reason for this behavior is that our current load balancing scheme is based entirely on the amount of work imposed by the backprojection computation. We are looking into ways of incorporating the projection data handling cost as well.

Before closing, we note that a tighter convex hull could be computed by allowing axial variation of its cross section, i.e., by using the individual Y projection boundary values as opposed to the minimum and maximum values only. The increase in computational cost associated therewith is small and most likely worth it. We are therefore looking into this issue as well. An important part of this work will be to develop a scheduling scheme that ensures good load balancing among the processors.

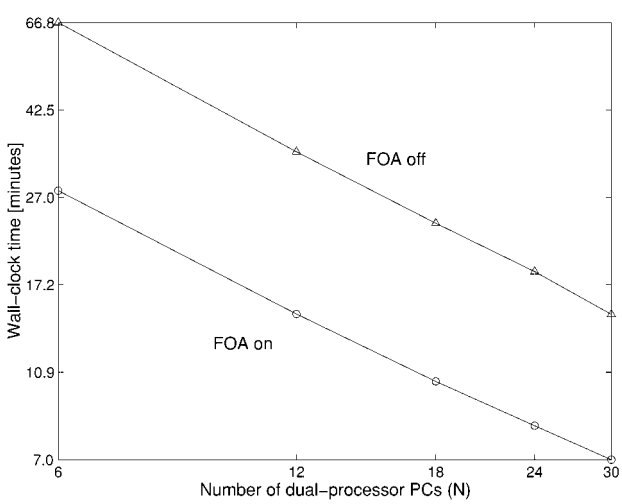

Figure 6. Wall-clock timing results (FOA = focus of attention).

Table I. Timing details w/FOA off.

N	Load/Weight/Filter	Backprojection
6	1.42	64.33
12	1.05	32.18
18	0.95	21.69
24	0.89	16.47
30	0.86	13.33

Table II. Timing details w/FOA on.

N	Load/FOA/Weight/Filter	Backprojection
6	1.57	25.78
12	1.23	12.97
18	1.10	8.75
24	1.02	6.64
30	1.11	5.38

ACKNOWLEDGMENTS

The mouse data was provided by Jurgen Seidel and Mike Green, both with the National Institutes of Health. The PC cluster was acquired as part of SInRG, a University of Tennessee grid infrastructure supported by the National Science Foundation under grant number EIA-9972889.

REFERENCES

L.A. Feldkamp, L.C. Davis, and J.W. Kress, Practical conebeam algorithm, J Opt Soc Am A 1 (1984), 612–619.

S.S. Gleason, H. Sari-Sarraf, M.J. Paulus, D.K. Johnson, S.J. Norton, and M.A. Abidi, Reconstruction of multi-energy x-ray computed tomography images of laboratory mice, IEEE Trans Nuc Sci 46 (1999), 1081–1086.

J. Gregor and D. Huff, A focus-of-attention preprocessing scheme for EM-ML PET reconstruction, IEEE Trans Med Imaging 16 (1997), 218–223.

J. Gregor and D. Huff, A computational study of the focus-of-attention EM-ML algorithm for PET reconstruction, Parallel Computing 24 (1998), 1481–1497.

J. Gregor and F.R. Rannou, A least squares framework for projection MRI reconstruction, Proc SPIE Vol 4322 pages (2001), 888–897.

J. Gregor and F.R. Rannou, 3D Support function estimation and application for projection MRI, Int J Imaging Syst Technol 12 (2002), 43–50.

A.C. Kak and M. Slaney, Principles of Computerized Tomography, SIAM, 2001.

W.C. Karl, S.R. Kulkarni, G.C. Verghese, and A.S. Willsky, Local tests for consistency of support hyperplane data, J Math Imaging and Vision 6 (1995), 249–267.

C. Laurent, F. Peyrin, J.M. Chassery, and M. Amiel, Parallel image reconstruction on MIMD computers for three-dimensional conebeam tomography, Parallel Computing 24 (1998), 1461–1479.

M.J. Paulus, S.S. Gleason, S.J. Kennel, P.R. Hunsicker, and D.K. Johnson, High resolution x-ray computed tomography: An emerging tool for small animal cancer research, Neoplasia 2 (2000), 62–70.

M.J. Paulus, H. Sari-Sarraf, S.S. Gleason, M. Bobrek, J.S. Hicks, D.K. Johnson, J.K. Behel, and L. Thompson, A new x-ray computed tomography system for laboratory mouse imaging, IEEE Trans Nuc Sci 46 (1999), 558–64.

J.L. Prince and A.S. Willsky, Reconstructing convex sets from support line measurement, IEEE Trans Pattern Anal Mach Intell 12 (1990), 377–389.

D. Reimann, C. Chaudhary, M.J. Flynn, and I. Sethi, Conebeam tomography using MPI on heterogeneous workstation clusters, Proc 2nd MPI Developers Conference and Users' Group Meeting pages 142–148, Univ Notra Dame, IN, 1996.

H. Turbell and P.E. Danielsson, Fast Feldkamp reconstruction, Proc Intl Meeting on Fully 3D Image Reconstruction in Radiology and Medicine (3D99) pages 311–314, Egmond aan Zee, The Netherlands, 1999.

G. Wang, S. Zhao, and P. Cheng, Exact and approximate conebeam x-ray microtomography, In P. Cheng, P. Huang, J. Wu, G. Wang, and H. Kim, editors, Focus of Multidimensional Microscopy volume 1, pages 233–261. World Scientific, 1999.