



## Journal Menu

- Abstracting and Indexing
- Aims and Scope
- Article Processing Charges
- Articles in Press
- Author Guidelines
- Bibliographic Information
- Contact Information
- Editorial Board
- Editorial Workflow
- Reviewers Acknowledgment
- Subscription Information

- Open Special Issues
- Published Special Issues
- Special Issue Guidelines

Call for Proposals for  
Special Issues

International Journal of Biomedical Imaging  
Volume 2007 (2007), Article ID 29160, 9 pages  
doi:10.1155/2007/29160

Research Article

## A Fast CT Reconstruction Scheme for a General Multi-Core PC

Kai Zeng,<sup>1</sup> Erwei Bai,<sup>2</sup> and Ge Wang<sup>3</sup>

<sup>1</sup>Biomedical Engineering Department, University of Iowa, Iowa City 52244, IA, USA

<sup>2</sup>Electrical and Computer Engineering Department, University of Iowa, Iowa City, 52244, IA, USA

<sup>3</sup>Biomedical Imaging Division, VT-WFU School of Biomedical Engineering and Sciences, Virginia Polytechnic Institute and State University, Blacksburg 24061, VA, USA

Received 28 February 2007; Accepted 24 April 2007

Academic Editor: Yibin Zheng

### Abstract

Expensive computational cost is a severe limitation in CT reconstruction for clinical applications that need real-time feedback. A primary example is bolus-chasing computed tomography (CT) angiography (BCA) that we have been developing for the past several years. To accelerate the reconstruction process using the filtered backprojection (FBP) method, specialized hardware or graphics cards can be used. However, specialized hardware is expensive and not flexible. The graphics processing unit (GPU) in a current graphic card can only reconstruct images in a reduced precision and is not easy to program. In this paper, an acceleration scheme is proposed based on a multi-core PC. In the proposed scheme, several techniques are integrated, including utilization of geometric symmetry, optimization of data structures, single-instruction multiple-data (SIMD) processing, multithreaded computation, and an Intel C++ compiler. Our scheme maintains the original precision and involves no data exchange between the GPU and CPU. The merits of our scheme are demonstrated in numerical experiments against the traditional implementation. Our scheme achieves a speedup of about 40, which can be further improved by several folds using the latest quad-core processors.

[Abstract](#)

[Full-Text PDF](#)

[Linked References](#)

[How to Cite this Article](#)