TSINGHUA SCIENCE AND TECHNOLOGY
ISSN 1007-0214 05/20 pp30-35
Volume 15, Number 1, February 2010

Fast Implementation of Iterative Reconstruction with Exact Ray-Driven Projector on GPUs

Fang Xu**

Siemens Corporate Research, Princeton, NJ 08540, USA

Abstract: Iterative methods are popular choices in image reconstruction fields due to their capability of recovering object information from incomplete acquisition data. However, the computation process involves frequent uses of forward and backward projections that are computationally expensive. Past research has proved that a forward projector that can produce high quality images is crucial to achieve a good convergence rate. In this paper a high performance iterative reconstruction framework is introduced, where two most popular iterative algorithms: Simultaneous Algebraic Reconstruction Technique (SART) and Ordered-subsets Expectation Maximization (OSEM) are supported. The framework utilizes Siddon's ray-driven method to generate forward projected images. Benefited from functionalities offered by current generation of graphics processing units (GPUs), it achieves better performance when compared to previous GPU implementations that use grid-interpolated methods, on top of the significant speedups over CPU-based solutions.

Key words: image reconstruction; tomography; graphics processing units

Introduction

Conventional analytical image reconstruction algorithms, such as filtered backprojection (FBP)[1], are simple and easy to implement. But they are not suitable for applications where there is a high degree of incompleteness of projection data, e.g., digital tomosynthesis[2] whose available images are generated from a series of limited viewing angles due to geometry constraints. To address the problem, iterative methods are often adopted due to their superior capability for solving ill-posed inverse problems. Previous studies have shown that they can not only produce results with higher in-depth resolution but effectively reduce ghost artifacts[3]. Among them, algebraic algorithms such as the simultaneous algebraic reconstruction technique (SART)[4], and statistical methods based on maximum-likelihood (ML), e.g., ordered-subset expectation maximization (OSEM)[5], are popular candidates. However, these iterative methods are computationally very intensive due to the massive dimension of system matrix that models the imaging process. Moreover, when coupled with projection images of high resolution generated from latest flat-panel detectors, the reconstruction process usually results in a very lengthy calculation time for CPU-based solutions. This has prevented iterative algorithms from being applied in time-critical clinical applications.

Recently graphics processing units (GPU) have emerged as a popular platform to perform numerous computationally intensive tasks thanks to its low-cost, commodity parallel-computing architecture. Various scientific applications including a wide area of medical imaging modalities have successfully utilized GPUs to boost their performance. Using GPUs for tomographic reconstruction is particularly attractive due to the

Received: 2009-10-18; revised: 2009-12-29
** To whom correspondence should be addressed.
E-mail: fang-xu@siemens.com

effective mapping of the algorithm and significant speedups it offers. Both analytical and iterative methods have been accelerated on GPUs, while competitive performance has been obtained compared to other popular platforms using parallel processors (Cell, FPGA and so on)[6-10]. The most significant speedups were observed on iterative applications due to their high complexity of computation. Specifically, forward projection, being the most frequently used component, has been extensively investigated. These include exact algorithms such as Siddon's method[11], where intersection lengths between a voxel and the ray are computed. For this, various acceleration schemes were proposed to make it amenable for parallel computing[12-14]. The other popular category is grid-interpolated method, where sample points are spaced in identical distance along the ray[15].

In spite of all the successful applications using GPUs, certain hardware limitations often prevent straightforward implementations of general-purpose computations due to the evolving programming interface. Before render-to-3-D-texture functionality is supported on the latest hardware, the reconstructed volume usually has to be represented as stacks of 2-D textures. Hence, approximate methods that result in loss of image quality have been used[10], or extra memory addressing schemes and management have to be introduced to implement the 3-D ray-driven forward projection[16]. To cope with the issue, here we designed a simplified GPU-based reconstruction framework that takes advantage of the latest functionalities to achieve optimal speedups without compromising the image quality.

1 Theory

The procedure of tomographic imaging can be modeled via the following equation:

$$p_i = \sum_{j=1}^{N} v_j \bullet w_{ij}, \quad i = 1, 2, \ldots, M \tag{1}$$

Here the pixel value p_i on the image plane is the line integral computed for the i-th ray that involves voxels v_j during the traversal throughout the volume. w_{ij} is the contribution factor from voxel v_j with respect to pixel p_i, while M and N refer to the total number of pixels and voxels. Iterative methods solve the above equation system by using a numerical approximation approach, where projections of current estimation volume are compared with ground truth (scanner images) to derive correction images, which are in turn backwardly projected to update the current volume estimation. By repeatedly performing the above steps the difference between the estimation and ground truth will reduce until convergence is reached. Equation (2) describes the SART algorithm adopted in this framework:

$$v_j^{k+1} = v_j^k + \lambda \frac{\sum\limits_{p_i \in P_\varphi} \frac{p_i - fp_i}{\text{sow}_i} w_{ij}}{\text{sow}_j} = v_j^k + \lambda \frac{\sum\limits_{p_i \in P_\varphi} \frac{p_i - \sum\limits_{l=1}^{N} v_l^k \bullet w_{il}}{\sum\limits_{l=1}^{N} w_{il}} w_{ij}}{\sum\limits_{p_i \in P_\varphi}^{N} w_{ij}} \tag{2}$$

Here, fp_i is the pixel value calculated from forward projection, sow_i and sow_j are the sum of weights of all voxels with respect to pixel p_i and voxel v_j with respect to all pixels on the projection. The algorithm requires three major components: forward projection of current estimated volume, computation of correction image between the scanner image and the forward projection, and lastly backward projection of the correction image to the estimated volume. Similarly, the OSEM algorithm can be formulated by

$$v_j^{k+1} = v_j^k \bullet \frac{\sum\limits_{p_i \in P_{\text{set}}} \frac{p_i}{fp_i} w_{ij}}{\text{sow}_j} = v_j^k \bullet \frac{\sum\limits_{p_i \in P_{\text{set}}} \frac{p_i}{\sum\limits_{l=1}^{N} v_l^k \bullet w_{il}}}{\sum\limits_{p_i \in P_{\text{set}}}^{N} w_{ij}} \tag{3}$$

These algorithms perform in such a way that they iterate through forward projection, correction computation, backward projection and volume update stages until a convergence is reached, which is usually defined using a threshold computed either between successively reconstructed volumes, or the forward projected images of the reconstructed volume and input scanner projections.

2 Implementation

2.1 Forward projection using Siddons' ray-driven projector

We implemented Siddon's forward projection method