Nuclear Instruments and Methods in Physics Research B xxx (2013) xxx–xxx

Contents lists available at SciVerse ScienceDirect

Nuclear Instruments and Methods in Physics Research B

journal homepage: www.elsevier.com/locate/nimb

Fast simulation of Proton Induced X-Ray Emission Tomography using CUDA

D.G. Beasley *, A.C. Marques, L.C. Alves, R.C. da Silva

Instituto Tecnológico e Nuclear, Instituto Superior Técnico, Universidade Técnica de Lisboa, E.N.10, 2686-953 Sacavém, Portugal
Centro de Física Nuclear da Universidade de Lisboa, Av. Prof. Gama Pinto 2, 1649-003 Lisboa, Portugal

ARTICLE INFO

Article history:
Received 23 July 2012
Received in revised form 17 December 2012
Accepted 18 December 2012
Available online xxxx

Keywords:
PIXE
STIM
Tomography
CUDA
Simulation

ABSTRACT

A new 3D Proton Induced X-Ray Emission Tomography (PIXE-T) and Scanning Transmission Ion Microscopy Tomography (STIM-T) simulation software has been developed in Java and uses NVIDIA™ Common Unified Device Architecture (CUDA) to calculate the X-ray attenuation for large detector areas. A challenge with PIXE-T is to get sufficient counts while retaining a small beam spot size. Therefore a high geometric efficiency is required. However, as the detector solid angle increases the calculations required for accurate reconstruction of the data increase substantially. To overcome this limitation, the CUDA parallel computing platform was used which enables general purpose programming of NVIDIA graphics processing units (GPUs) to perform computations traditionally handled by the central processing unit (CPU). For simulation performance evaluation, the results of a CPU- and a CUDA-based simulation of a phantom are presented. Furthermore, a comparison with the simulation code in the PIXE-Tomography reconstruction software DISRA (A. Sakellariou, D.N. Jamieson, G.J.F. Legge, 2001) is also shown. Compared to a CPU implementation, the CUDA based simulation is approximately 30× faster.

1. Introduction

Proton Induced X-Ray Emission Tomography (PIXE-T) has been in development for over 20 years [1] and has recently been performed at the Instituto Tecnológico e Nuclear, Instituto Superior Técnico (ITN-IST). Combined with Scanning Transmission Ion Microscopy-Tomography (STIM-T) and Rutherford Backscattering Spectroscopy, it can produce quantitative 3D elemental maps. Biological samples have been analysed [2,3] although the success has been limited for trace elements due to the significant counts required to reconstruct accurate tomograms, which is also hindered by the sample sensitivity to beam radiation damage [4]. Multiple detectors can be used to increase the geometric efficiency. However, as the detector solid angle increases the calculations required to accurately reconstruct the data increase substantially. While Filtered Back Projection (FBP) is suitable for STIM-T, iterative techniques are required to reconstruct elements from PIXE-T data due to the X-ray attenuation and the non-linear X-ray production. The most complete software is the Discrete Image Space Reconstruction Algorithm (DISRA) developed by Sakellariou [5], which simulates PIXE-T and the X-ray attenuation by segmenting the X-ray detector, and determining the attenuation from each voxel to every segment (detector pixel) for every projection and for each of the characteristic X-ray of interest. It is a computationally demanding simulation and this is performed through several iterations.

To improve planning of experiments (e.g. estimating count rate), and to accelerate the reconstruction of data, a new PIXE-T and STIM-T simulation program has been written in Java and CUDA. Common Unified Device Architecture (CUDA) is developed by NVIDIA™ for enabling general purpose computing of graphics processing units (GPGPUs) in order to take advantage of the massive parallel computing power of NVIDIA™ graphics cards. GPUs contain many more cores than central computer units (CPUs), although of a more simple type. For certain calculations GPUs can increase the speed considerably by performing a large number of tasks at the same time. In recent years many GPGPU implementations of X-ray-casting have been developed for use with computed tomography (CT) [6,7]. PIXE-T is an emission-based tomographic technique with a cone-beam geometry, similar to Positron Emission Tomography (PET) and Single Photon Emission Computer Tomography (SPECT) [8]. The requirements in terms of memory storage and parallelising differ substantially and are discussed below.

There are several methods, called projectors, for calculating the attenuation between two points in a voxel-based phantom. Variations of Siddońs algorithm [9] are often used with CPUs, where the length of the intersection of the ray through each voxel is calculated. However this is not an efficient method using GPUs due to the use of conditions [10]. The trilinear projection method is more computer intensive using CPUs but can be implemented on GPUs

---

* Corresponding author at: Instituto Tecnológico e Nuclear, Instituto Superior Técnico, Universidade Técnica de Lisboa, E.N.10, 2686-953 Sacavém, Portugal. Tel.: +351 966011864.
E-mail address: dgbeasley@itn.pt (D.G. Beasley).

0168-583X/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.nimb.2012.12.053

Please cite this article in press as: D.G. Beasley et al., Fast simulation of Proton Induced X-Ray Emission Tomography using CUDA, Nucl. Instr. Meth. B (2013), http://dx.doi.org/10.1016/j.nimb.2012.12.053

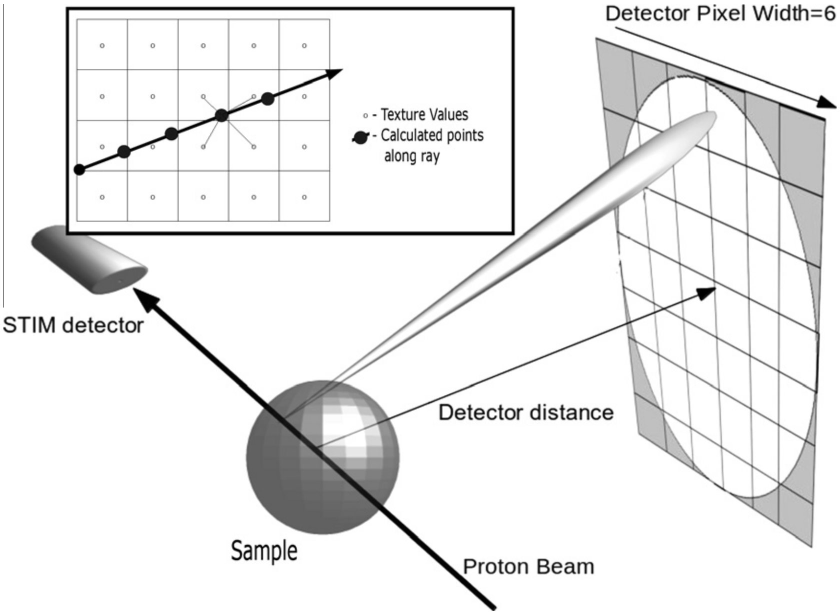

Fig. 1. Illustration showing the X-ray attenuation path (ray) of X-rays induced by the proton beam and the simulated segmentation of the detector. Inset: Illustration of interpolation of CUDA texture memory along the ray, showing the boundaries of the voxels, the texture value positions and X-ray path and positions where the attenuation is calculated. The contribution from the surrounding texture values are weighted by the distance from the calculated value.

very efficiently. Graphics cards are designed specifically to calculate how images are displayed and are constructed with specific types of memory to accelerate this process. Texture memory is a type of memory that stores images as 4D vectors. Arrays of data are mapped to texture memory (therefore assigning values to points in this image space) and values can then be retrieved by sampling from this memory through interpolation, i.e. the value is derived from the weighted average of the surrounding mapped points, as illustrated in Fig. 1 in 2D form. It is very fast to access from the GPU and hence trilinear projection methods are very efficiently implemented. An example of trilinear interpolation projection called "line-of-sight" is provided by the CUDA Software Development Kit (SDK) in C++ and can be adapted for CT [6]. However, this is for producing an image from an external point source; for PIXE the X-rays are generated from within each voxel and, unlike CT, the sum of the projected image from each voxel for multiple characteristic energies is calculated.

A CUDA kernel, computer code compiled and executed on the graphics card, was written for calculating the attenuation correction factors, i.e. the attenuation paths from every voxel to every segment of a detector for every characteristic energy of interest. The JCuda library [11] was used to allow CUDA to be executed by the Java-written simulation.

2. Methods

2.1. Description of the simulation program

2.1.1. Overview

Apart from the X-ray attenuation calculations, the program is written primarily for simplicity and is not optimised for speed. As a simulation without considering X-ray attenuation takes an order of 2–3 s to perform, it is not deemed necessary.

Presented with a graphical user interface (GUI), the user can choose a phantom from a list or select real data from which a basic phantom is created based upon a FBP reconstruction. STIM projection data is used to detect the edge of the sample in order to create a mask that is applied to reconstructed data. A CPU or CUDA simulation can be selected via the GUI. For elements present in the sample, cross-section look-up tables are generated for X-ray production, proton stopping powers and X-ray attenuation data. These are read in from an Excel spread sheet book that can be easily modified. In the current form, the data tables used are the same as the DISRA package with the addition of stopping powers updated from SRIM [12]. These include elemental and compound stopping powers for proton energies below 1 MeV. Due to the tomography set-up at ITN, for many samples it is expected the residual proton energy will be below 1 MeV where chemical bonding may be affective. The compound stopping powers can be used for a-priori matrix composition, i.e. the bulk of the sample if it cannot be detected using PIXE (e.g. in light element matrixes).

The X-ray transmission values per μm are calculated for every voxel and stored in a 3D array. Before simulating a projection, the array is used to determine the total X-ray attenuation values for each X-ray line in every voxel. The edges of the sample are used to trim the 3D X-ray transmission arrays and to remove empty space, which in the case of CUDA is done before transferring the data to the graphics card in the form of CUDA arrays. Although CUDA offers massively parallel computing, the largest bottleneck is the transferring of data to and from the graphics cards memory (device) to the computer (host). Reducing the size of the CUDA arrays reduces transfer times considerably as well as the interpolation time.

2.1.2. CPU implementation

For the CPU implementation a very simple (and not very accurate) method written in Java is employed for X-ray attenuation calculations. Prior to the simulation, the incremental paths of the X-ray for each projection are stored in arrays. The ray is traced in steps equal to the width of the voxels, and the attenuation accumulated until the edge of the phantom.

2.1.3. CUDA implementation

For an X-ray of a particular energy, the total X-ray yield, $Y(x,y)$ per scanned pixel with given coordinates x and y is given by the sum of the contributions along the length of the beam in the z-direction:

Please cite this article in press as: D.G. Beasley et al., Fast simulation of Proton Induced X-Ray Emission Tomography using CUDA, Nucl. Instr. Meth. B (2013), http://dx.doi.org/10.1016/j.nimb.2012.12.053