General Purpose Graphics Processing Unit Speedup of Integral Relative Electron Density Calculation for Proton Computed Tomography

S. A. McAllister, K. E. Schubert, R. Schulte, S. Penfold

I. Introduction

CLINICAL application of protons was first suggested over 60 years ago [9]. Proton radiation can deliver high doses of radiation to tumors or other targets close to critical structures, and thus is vitally important for modern 3D conformal radiation therapy. Currently proton dose calculations rely on x-ray computed tomography (CT), which limits their accuracy due to the physical interaction differences of protons and x-rays. To gain the maximum benefit from proton therapy, proton computed tomography (pCT) offers the opportunity to more accurately plan proton doses and to verify the correct proton beam delivery in the treatment position. This is accomplished by choosing the proton energy sufficiently high to penetrate the patient and by reconstructing density values based on energy loss measurements [1]. As an additional advantage, pCT achieves similar density resolution with lower dose than x-ray CT, because each proton is tracked individually. Despite these advantages, a fully operational pCT system does currently not exist, in part, related to the large amount of proton and object data that need to be acquired and reconstructed, respectively. Preliminary work in proton CT over the last several years has centered on the most likely path formalism [8], image reconstruction [6], [3], [4], [5], and basic design of a system [7]. A vital step in the reconstruction of a pCT image is the calculation of the integral relative electron densities, which must be known for each of the proton histories. The number of histories can run into the hundreds of millions. Efficient calculation is important if the overall clinical goal of a pCT reconstruction in 5 minutes is to be achieved. Figure 1 shows the complete path of the proton histories through the image reconstruction process. This paper will focus on the hardware acceleration of the integral relative electron density calculation. The integral relative electron density requires the incoming and outgoing energies of the protons being tracked, which is given by the detectors. These values are not used in the iterative reconstruction process for reconstructing the image. Rather, they are calculated before the reconstruction iterations and stored with their corresponding proton history data.

Manuscript received July 25, 2009. The support of NIH under grant #HD052368 is gratefully acknowledged. S. A. McAllister (smcallis@csusb.edu) and K. E. Schubert (schubert@csusb.edu) are from the Department of Computer Science and Engineering, California State University, San Bernardino, 5500 University Parkway, San Bernardino, CA, USA 92407. R. Schulte (rschulte@dominion.llumc.edu) is from the Department of Radiation Medicine, Loma Linda University Medical Center, 11234 Anderson St., Loma Linda, CA, USA 92350. S. Penfold (snp75@uow.edu.au) is from the Centre for Medical Radiation Physics, University of Wollongong, Wollongong, NSW 2522, Australia.

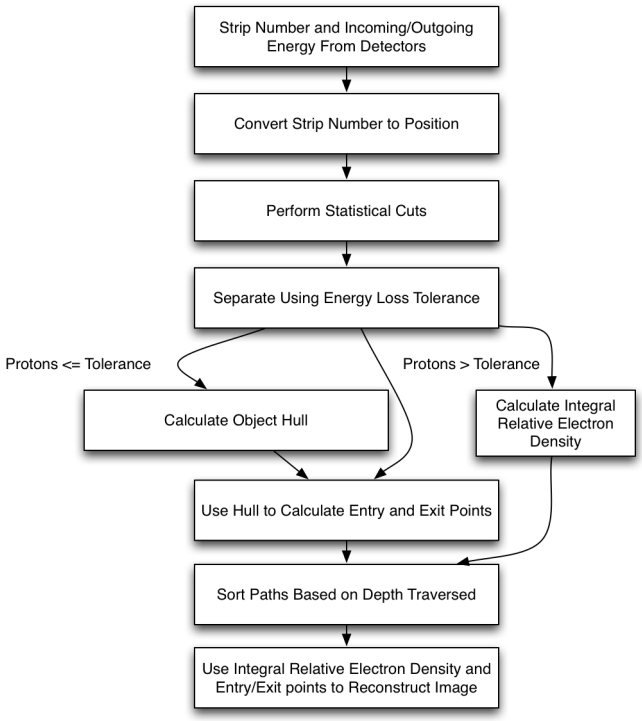

Fig. 1. The path of the data as it passes through the image reconstruction process.

II. Bethe-Bloch Equation

As the proton paths traverse an object, they deviate from a straight line due to the effects of multiple Coulomb scattering (MCS) and must be tracked using a formalism that models MCS [8]. A sparse iterative solver, like the algebraic reconstruction technique (ART), is used to reconstruct the object using these non-linear paths [6]. An example of a reconstructed phantom can be found in Figure 2. This reconstruction was obtained from [6]. In order to perform the reconstruction, the path integral of relative electron density along the most likely proton path must be calculated. This is accomplished by calculating the integral given by

$$\int_{E_{out}}^{E_{in}} \frac{dE}{F(E, I_{water})}$$

Fig. 2. Phantom reconstructed using a variation of the algebraic reconstruction technique.

The function $F(E, I_{water})$ can explicitly be expressed as

$$F(I, E(U)) = K\frac{1}{\beta^2(u)}\left[\ln\left(\frac{2m_ec^2}{I_{water}}\frac{\beta^2(u)}{1-\beta^2(u)}\right) - \beta^2(u)\right]$$

where m_ec^2 is the electron rest energy $(511.011KeV)$, and $\beta(u)$ is the proton velocity at depth u relative to the speed of light c. The constant K is defined as

$$K = 4\pi r_e^2 m_e c^2 \approx 0.170\frac{MeV}{cm}$$

where r_e is the classical electron radius $(2.818 \times 10^{-13}cm)$. The relativistic relationship between β and E is given by

$$\beta(u) = \sqrt{1 - \left(\frac{E_p}{E(u) + E_p}\right)^2}$$

where E_p is the proton rest energy $(938.272MeV)$.

Since the depth dependence of I is usually not exactly known (because the object composition is not a priori known), integration of this equation is only possible by assuming a reasonable approximation of I. For human tissues encountered in proton CT, the variation of I is not very large, and the function F has only a weak logarithmic dependence on I. Therefore, it is reasonable to use the value of the mean excitation potential of water, which is 75.0 eV. Also note that the formula given here is only an approximation of the original Bethe Bloch equation, which contains a term W_{max}, the maximum energy transfer in a single collision. The approximation here is valid if the mass of the incident particle is large relative to the electron mass, which is the case for protons ($m_p/m_e \approx 1800$).

The integrand was simplified for implementation on either a multi-core processor or a GPGPU by showing

$$\begin{aligned} F(I, E(U)) &= K\left(1 + \frac{E_p^2}{E^2 + 2EE_p}\right)\left[\ln\left(\frac{2m_ec^2}{I_{water}}\right) + \right. \\ &\quad \ln(E) + \ln(E + 2E_p) - 2\ln(E_p) - 1 + \\ &\quad \left.\frac{E_p^2}{(E + E_p)^2}\right] \end{aligned}$$

The integrals were then evaluated using 2 point Gaussian quadrature for every 0.125mm of depth traversed.

III. GPGPU Programming and Results

A major impediment to efficient GPGPU code is the tendency of programmers to write thunks, which are computations that return results, see [2]. On a GPGPU a thunk causes the GPGPU to pass back intermediate results to the host, which then must send them back for further computation, causing two bus transfers per thunk, which are often in a loop and thus greatly magnify the negative effects of the bus transfers. In particular thunking causes the GPGPU computation to be limited by the bus transfer, and thus the speedup becomes limited due to Amdahl's Law:

$$speedup = \frac{1}{f + \frac{1-f}{2*cores}}.$$

Note that speedup of the parallel portion is twice the number of cores because GPGPUs do two floating point operations, multiply and add, at a time. Typical thunking code on a 240 stream processors (such as the GTX280 used) spends about 2% ($f = 0.02$) or more of its time in memory transfers. Thunking code is thus limited to a speedup of less than 50, which corresponds to an efficiency of around 10%. To prevent thunking the input and output energies are copied to the GPU as floats and nothing is returned until the results are copied back, again as floats. Intermediate results are stored in dynamically allocated arrays on the GPGPU, which are explicitly created before the calculation begins, and then freed at the conclusion of the computation. This makes the computation very fast, but memory limited, so that approximately 24 million histories can be handled per 1 GB of GPGPU RAM.

Shown in Table I are CPU and GPU times for the integral relative electron density equation. Times are in milliseconds and are averages of times for a given number of elements over 1000 iterations. It also shows that before one million proton energies the GPU does not have much, if any, advantage over the CPU. The reason for this is the GPU initialization time. After six million elements the GPU ran out of memory and returned no values. With batches of five million proton energies, 100 million proton energies could be calculated in 77.12ms while on a CPU, 100 million proton energies would take 38.39 seconds to calculate.

The CPU used in the comparison was an Intel Core 2 Quad Q6600 2.4 GHz on an ASUS P5N32-E SLI LGA 775 motherboard with 4 GB of Corsair Dominator RAM at 800 MHz. The GPU used was an NVIDIA GTX280, which contains 240 processing cores running at 1296 Mhz and 1

Number of Elements	CPU	GPU	Speedup
1×10^2	0.034	0.100	0.3400
1×10^3	0.329	0.102	3.2255
1×10^4	3.803	0.105	36.219
1×10^5	38.131	0.185	206.11
1×10^6	384.776	0.860	447.41
2×10^6	767.827	1.423	539.58
3×10^6	1205.037	2.564	469.98
4×10^6	1527.700	3.057	499.74
5×10^6	1919.564	3.856	497.81
6×10^6	2293.294	4.550	504.02
7×10^6	3666.551	N/A	N/A

TABLE I
CALCULATION TIME OF INTEGRAL RELATIVE ELECTRON DENSITY CALCULATION IN MILLISECONDS.

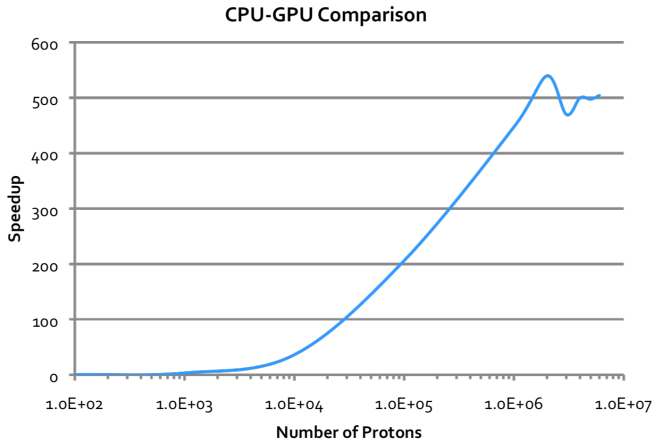

Fig. 3. Graphicss processing unit speedup of integral relative electron density calculation.

GB of RAM. The GPGPU was programmed using a subset of C called CUDA (Compute Unified Device Architecture). Because GPUs are accessed via a PCI Express 2.0 bus, they can handle up to 8 GB/s in and 8 GB/s out (500 MB/s * 16 lanes). In the case of pCT, the PCI Express bus is a bottleneck because memory bandwidths on the motherboard as well as the GPU are much faster. Because of this limitation, algorithms need to be designed to maximize the number of calculations per data transfer.

CPU and GPU times for the integral relative electron density equation were averaged over 1000 iterations; the results are shown in Figure 3. As already described, before ten thousand proton histories, the GPU does not have any advantage over the CPU. This is because of the GPU initialization time and the reduced efficiency of transferring small blocks. After one million iterations the speedup reached a maximum of around 500 times over the CPU; however, after six million elements the GPU ran out of memory and returned no values. Newer GPGPUs have up to 4 GB of RAM, so we expect they can handle up to 25 million proton energy calculations before running out of memory, thus achieving an even larger speedup. The next generation of Nvidia GPGPUs, code named Fermi, will have twice as many CUDA cores, and is capable of handling 256 double precision floating point calculations per clock cycle, which is very important for reducing errors. The next generation also has cache and ECC memory (error correction), which will improve speed and reliability. Even at the current optimum of batches of five million proton histories, 100 million relative electron density integrals can be calculated in under a second, which is a significant improvement over the 8+ minutes a cpu would take.

IV. CONCLUSIONS

The significant speedup of the GPGPU calculations make the crucial step of calculating the integral relative electron density feasible in a clinical setting. These numbers also suggest there could be a significant improvement of the iterative component of the reconstruction as well. Furthermore, this work has suggested the possibility of binning protons by their integral relative electron densities, which would lead to a drastic reduction in memory transfers to and from the GPGPU, leading to significant improvements in reconstruction time. The energy binning technique is currently being researched and will be presented at a future date.

REFERENCES

[1] K. M. Hanson, J. N. Bradbury, T. M. Cannon, R. L. Hutson, D. B. Laubacher, R. J. Macek, M. A. Paciotti, and C. A. Taylor. Computed tomography using proton energy loss. *Physics in Medicine and Biology*, 26:965–983, November 1981.

[2] P.Z. Ingerman. Thunks - A Way of Compling Procedure Statements with Some Comments on Procedure Declarations. *IFIP ALGOL Bulletin & CACM*, 1(1):55–58, January 1961.

[3] T. Li, Z. Liang, K. Mueller, J. Heimann, L. Johnson, H. Sadrozinski, A. Seiden, D. Williams, L. Zhang, S. Peggs, T. Satogata, V. Bashkirov, and R. Schulte. Reconstruction for Proton Computed Tomography: A Monte Carlo Study. In *IEEE Nuclear Science Symposium Conference Record*, volume 4, pages 2767–2770, October 2003.

[4] T. Li, J. Singanallur, T. Satogata, D. Williams, and R. Schulte. Reconstruction for Proton Computed Tomography by Tracing Proton Trajectories: A Monte Carlo Study. *Med Phys*, (33):699–706, March 2006.

[5] K. Mueller, Z. Liang, T. Li, F. Xu, J. Heimann, L. Johnson, H. Sadrozinski, A. Seiden, D. Williams, L. Zhang, S. Peggs, T. Satogata, V. Bashkirov, and R. Schulte. Reconstruction for Proton Computed Tomography: A Practical Approach. In *IEEE Nuclear Science Symposium Conference Record*, volume 5, pages 3223–3225, October 2003.

[6] S. N. Penfold, R. W. Schulte, Y. Censor, V. Bashkirov, S. McAllister, K. E. Schubert, and A. B. Rozenfeld. Block-Iterative and String-Averaging Projection Algorithms in Proton Computed Tomography Image Reconstruction. In *The Huangguoshu International Interdisciplinary Conference on Biomedical Mathematics: Promising Directions in Imaging, Therapy Planning and Inverse Problems*, in press, 2009.

[7] R. Schulte, V. Bashkirov, T. Li, J. Z. Liang, K. Mueller, J. Heimann, L. R. Johnson, B. Keeney, H. Sadrozinski, A. Seiden, D. C. Williams, L. Zhang, Z. Li, S. Peggs, T. Satogata, and C. Woody. Design of a Proton Computed Tomography System for Applications in Proton Radiation Therapy. *IEEE Transaction on Nuclear Science*, 51(3):866–872, June 2004.

[8] R. W. Schulte, S. N. Penfold, J. E. Tafas, and K. E. Schubert. A maximum likelihood proton path formalism for application in proton computed tomography. *Med Phys.*, (35):4849–4856, November 2008.

[9] R. R. Wilson. Radiological Use of Fast Protons. *Radiology*, (47):487–491, 1946.