

Accelerated cone-beam backprojection using GPU-CPU hardware

Dmitri Riabkov, Xinwei Xue, Dave Tubbs, Arvi Cheryauka

Abstract— The three-dimensional image reconstruction process used in interventional CT imaging is computationally demanding. Implementation on general-purpose computational platforms requires substantial processing time, which is undesirable during time-critical surgical and minimally invasive procedures. Central and Graphics Processing Units (CPUs and GPUs, respectively) have been studied as a platform to accelerate 3-D imaging. GPU devices offer a programmable hardware architecture, suitable for pipelining and high levels of parallel processing to increase computational throughput, as well as the benefits of being off-the-shelf and effectively scalable solutions. The focus of this paper is on the backprojection step of the image reconstruction process, since it is the most computationally intensive part. Using the modified Feldkamp-Davis-Kress (FDK) cone-beam algorithm, our feasibility studies indicate the entire 512^3 image reconstruction on a mobile X-ray C-arm can be accelerated to real time (i.e. completed immediately after an exposure scan of 15-30 seconds duration).

Index Terms—X-ray tomography, image reconstruction

I. INTRODUCTION

CT imaging using a mobile X-ray C-arm system in interventional and minimally-invasive surgery requires true- or near- real time computing solutions that meets ‘performance-per-watt’ and ‘performance-per-dollar’ constraints (Fig. 1) [6]. The computational requirements are increasing rapidly, mainly due to the need for rapid access to medical imagery at any time before, during or after the procedure.

The use of off-the-shelf commodity computers is attractive, but is not always power and cost efficient. On the other hand, the highly parallel nature of Radon transform and CT algorithms enable embedded parallel computing to gain a significant boost of performance while the power budget remains manageable from a single wall outlet.

The purpose of this paper is to look at the ‘pros’ and ‘cons’ of 3-D GPU accelerated image reconstruction. In particular, we examine the design and implementation capabilities of performing cone-beam backprojection following the X-ray projection rate to enable delivery of the 3-D volume immediately after (so-called real-time) or shortly after scan acquisition is completed.

D. Riabkov, X. Xue, D.Tubbs, A. Cheryauka are with General Electric Healthcare – Surgery, Salt Lake City, UT, 84116, USA (phone: 801-536-4529, fax: 801-535-4816; e-mail: Dmitri.Riabkov@ge.com).



Fig. 1. GE-OEC 9900 Elite Mobile X-ray C-arm system for interventional and minimally-invasive surgery.

II. CONE BEAM BACKPROJECTION

In our studies, we implement cone beam image reconstruction utilizing a modified FDK algorithm [5]. Correction, rebinning, weighting, and filtering of the projection data are fast operations, where their execution times take only a few percent of the total volume backprojection time. Our focus is the feasibility of completing backprojection computations using existing C-arm workstation hardware within the C-arm CT scan.

It is commonly known that backprojection restores the attenuation value at each volumetric element of the Region-Of-Interest (ROI). Inversion formula in the form of filtered backprojection and FDK-type approximation for irregular circular acquisition is described as follows:

$$p(\mathbf{r}) = \int_{\Lambda} d\lambda \omega^2(\lambda, \mathbf{r}) g_F(u(\lambda, \mathbf{r}), v(\lambda, \mathbf{r})), \quad \mathbf{r}(x, y, z) \in R^3, \quad (1)$$

$$\omega[u, v, 1]^T = \mathbf{M}(\lambda) \cdot [x, y, z, 1]^T.$$

Here p is the reconstructed volumetric data, g_F is the filtered projection data, Λ is the scanning angular interval, ω is the distance weight, u is the detector axial coordinate, v is the detector longitudinal coordinate, and \mathbf{M} is a 3x4 projection matrix. The projection matrix \mathbf{M} is the combined result of C-gantry intrinsic and extrinsic geometry calibrations [2-3].

Under cone-beam geometry, accumulation of filtered contributions from the rays passing a cubic volume has computational complexity of $O(N_{view} * N_{voxel}^3)$, where N_{view} and N_{voxel} are the numbers of projection views and voxels in the ROI in one dimension, respectively. In other words, to backproject the volume using plain projections, one needs to program, generally, 4 nested loops.

III. IMPLEMENTATION

In our studies, GPU-CPU hardware was targeted to perform 512-cubed backprojection. Reconstruction of a set of 512²-resolution slices is currently a standard in industrial X-ray CT.

It can be seen from the publications that GPUs are widely used for general purpose computing, beyond the original target of computer graphics and gaming industry [8]. The earliest attempt to accelerate CT reconstruction using graphics hardware dates back to 1994, when Calbral *et al.* utilized the texture mapping hardware for 3-D reconstruction [1]. With the introduction of modern GPUs in the last 5 years, both analytical and iterative methods have been implemented on graphics hardware [9-12, 14-17].

A. Architecture & GPGPU Programming Model

The GPU pipeline has programmable vertex and fragment processors, which enables the processing of multiple data-parallel primitives. The details on GPU architecture and general-purpose GPU programming model can be found, for instance, in [7,14].

B. Algorithm Design

Our voxel-driven implementation follows the steps in Fig. 2 and Fig. 3. Two governing scenarios comprising different loop structure were considered. The general idea is that an outer loop resides on the CPU, while an inner loop resides on the GPU. We implemented slice-wise and projection-wise approaches to estimate the timing and memory requirements:

In the slice-wise case (Fig. 2), the projections (views) reside in GPU memory, and each volume slice is reconstructed in the GPU. After each GPU projection loop is done, the reconstructed slice is uploaded to the CPU memory.

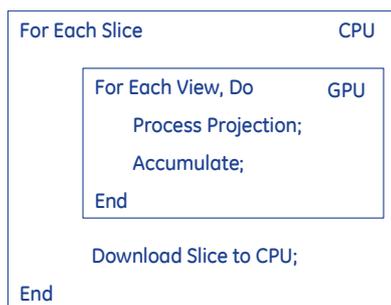


Fig. 2. Slice-wise backprojection.

In the projection-wise case (Fig. 3), the reconstructed volume resides in GPU texture memory, while each projection is uploaded to the GPU memory dynamically.

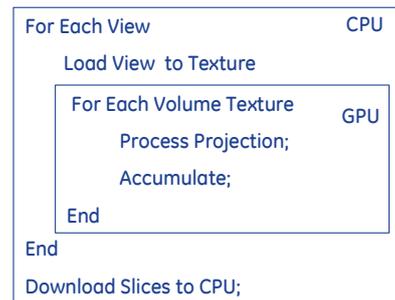


Fig. 3 Projection-wise backprojection

Both techniques use offscreen buffer (pbuffer) and frame buffer objects (FBO) supported by OpenGL 2.0 and modern GPUs.

These two approaches make an assumption that either all the projections or the reconstructed volume could fit in the GPU texture memory.

The advantage of the slice-wise implementation is that it is simple and fast, without overheads of context switching. Due to a limited amount of GPU memory, the projections are limited in size and number. The computation can only start after all the projections are acquired and reside in GPU memory.

The projection-wise approach has no limitations on the input projection data as long as the reconstruction volume can be stored in the GPU memory. So, the calculations can be performed during the CT scan, starting from the moment the first projection is acquired.

C. Experimental Setup

Two PC configurations comprising the following off-the-shelf components were exercised:

- Intel P4 XE, 3.4 GHz, 4GB RAM, 800 MHz FSB, Nvidia Quadro FX4500 w/ 512MB, SUSE Linux 10.0, 32-bit
- Intel dual-core Xeon (Woodcrest), 3.0 GHz, 8GB memory, 1333 MHz FSB, Nvidia GeForce 8800 GTX w/ 768MB, SUSE Linux 10.1, 64-bit

The Quadro FX4500 belongs to the professional card category, and is targeted for graphics workstations. The GeForce 8800 GTX is the latest consumer card from Nvidia, and is built on a unified G80 architecture with utilization of CUDA technology [13]. Its theoretical peak performance can be estimated as large as 345.6 Gflops, i.e. 128 (streaming processors) \times 1350 MHz (shader clock rate) \times 2 flops (floating point operations per clock).

We have implemented a 512³ volume backprojection, processing a series of 1024² plane projection data. While

both the input projections and the final reconstructed volume are stored in 16-bit integer format, all internal calculations are performed in 32-bit floating point format. Signal restoration in between pixel space is done by using hardware-based bi-linear interpolation (LI).

D. Synthetic Results

Synthetic data are generated with use of 3-D Shepp-Logan phantom. It's dimensionless geometry and attenuation were scaled to the size of a human head, and 'water-to-bone' values of 75 KeV, were used. Noisy data were weighted and filtered in advance. The results of the GPU backprojection, shown in Fig.4, are achieved with accuracy practically identical to accuracy of the CPU-based backprojection results (32-bit floating point). For benchmarking purposes, 165 input projections were used.

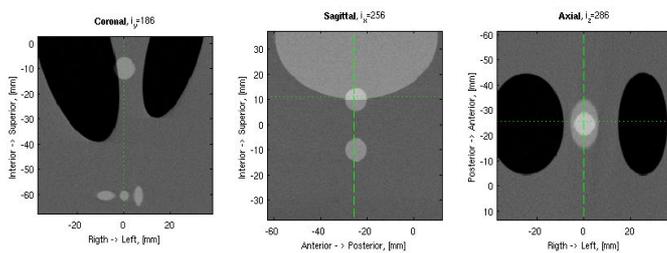


Fig. 4 ROI reconstruction of $\sim 80^3$ mm volume.

The memory allocation and execution time data for the slice-wise approach are given in Table 1.

TABLE 1: SLICE-WISE BACKPROJECTION

| GPU / CPU | GPU memory (MB) | Time (sec) |
|-------------------------|-----------------|------------|
| Quadro FX 4500 / P4 | 347 | 20.7 |
| GeForce 8800 GTX / Xeon | | 2.7 |

The projection-wise data are shown in Table 2.

TABLE 2: PROJECTION-WISE BACKPROJECTION

| GPU / CPU | GPU memory (MB) | Time (sec) |
|-------------------------|-----------------|------------|
| Quadro FX 4500 / P4 | 539 | N/A |
| GeForce 8800 GTX / Xeon | | 5.05 |

Note: since the 32-bit 512^3 volume does not fit in 512MB memory on the Quadro FX 4500, the calculation results are

not available. Bit conversion and data transfer from GPU texture to CPU memory takes 1.42 seconds and is not included in the total time.

E. Experimental Results

Experimental data has been acquired with use of programmable rotating stage. The assembly of the cone-beam X-ray tube and the detector is stationary while anatomy-mimicking objects move along a programmable trajectory. The imagery results of knee and hand anthropomorphic phantoms are shown in Figures 5 and 6, respectively.

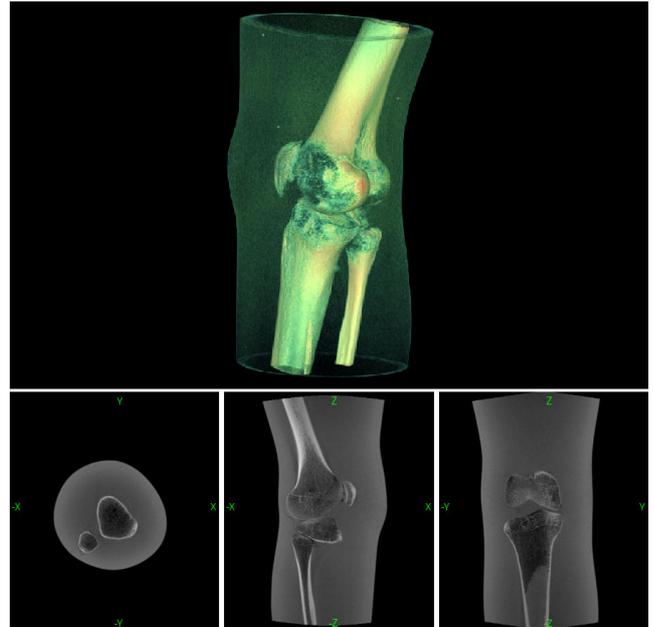


Fig. 5 Reconstruction of the knee phantom: the orthogonal views and volume rendering.

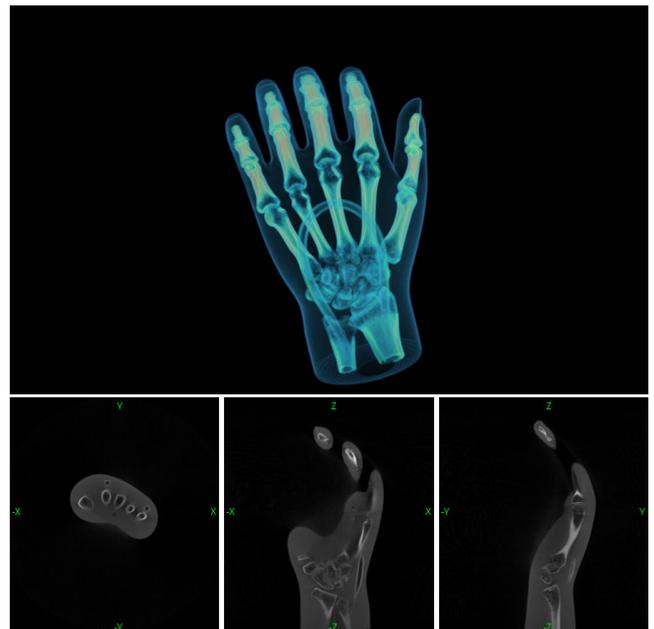


Fig. 6 Reconstruction of the hand phantom: the orthogonal views and volume rendering.

F. Other Results

A comprehensive review of speed up efforts in CT image reconstruction done by other groups has been recently provided by Kachelrieß *et al.* [9]. For purposes of quantitative comparison the results were scaled to the computational setup for cone-beam backprojection with the following characteristics: 512^3 output volume, 512 input projections, and 1 unit running a 3 GHz frequency process. Our projection-wise results obtained on the GeForce 8800 GTX / Xeon under the proposed above parameters would complement the data there in the ‘direct’ category as follows (Table 3):

TABLE 3: BACKPROJECTION PERFORMANCE. ALL VALUES HAVE BEEN SCALED TO 512 PROJECTIONS AND 512^3 VOXELS. ALL VALUES WERE FURTHER SCALED TO A SINGLE UNIT, I.E. TO ONE CPU, ONE FIELD PROGRAMMABLE GATE ARRAY (FPGA), ONE GPU, AND TO ONE CELL BROADBAND ENGINE (CBE), RESPECTIVELY, AND TO 3.0 GHZ IN THE CASE OF CPU AND CBE-BASED ALGORITHMS. THE TYPE COLUMN SPECIFIES THE INTERPOLATION TYPE, NEAREST NEIGHBOR (NN) OR BILINEAR INTERPOLATION (LI) AND THE TYPE OF ARITHMETICS USED: F+NUMBER OF BITS DENOTES FLOATING POINT ARITHMETICS WHILE I+NUMBER OF BITS STANDS FOR INTEGER (FIXED POINT) ARITHMETICS. THE TIME T PER 512×512 SLICE IS MEASURED AS AN AVERAGE OF THE TIME 512 T REQUIRED TO BACKPROJECT THE WHOLE VOLUME.

| | Type | HW | T ⁻¹ (fps) | T (ms) | 512 T (sec) | Notes |
|---------------------------|------------|-------------|--------------------------|-----------|----------------|--------|
| ... | ... | ... | ... | ... | ... | ... |
| Kachelrieß et al [9] | LI/ F32 | CBE | 18.8 | 53.1 | 27.2 | Direct |
| Riabkov ^[this] | LI/ F32 | GPU- CPU | 40.16 | 24.9 | 12.8 | Direct |
| ... | ... | ... | ... | ... | ... | ... |

To achieve higher performance, it is critical to restructure the programs to utilize internal unit instructions and memory bandwidth efficiently [9-10].

IV. CONCLUSION

The results of our efforts have shown that ‘on-the-fly’ 3D reconstruction on a mobile X-ray C-arm during the acquisition sweeps can be accomplished. The image quality required by interventional procedures and obtained by porting the reconstruction algorithms into GPU-CPU computational engines is confirmed. It is also important to maintain hardware compatibility and configurable codes. The examined heterogeneous platforms enable fast backprojection at an acceptable level of programming effort. Bringing computing capabilities to a real-time scenario would allow medical imaging to move to a more advanced state that may lead to a higher spatial/contrast/temporal resolution, higher accuracy, use of advanced algorithms, or new conceptual designs on clinical floor.

V. ACKNOWLEDGMENT

We are thankful Dmitry Samsonov (GE Healthcare, Haifa), Stephen Zingelewicz (GE Global Research, Niskayuna), and Francois Falco (GE Healthcare, Salt Lake City) for their helpful insights. We are also grateful to Intel Corporation for providing the high-end workstations used in these feasibility studies.

REFERENCES

- [1] Cabral B., Cam N., and J. Foran, 1994, “Accelerated volume rendering and tomographic reconstruction using texture mapping hardware”, *Proceedings of Symp. on Volume Visualization*, pp. 91-98.
- [2] Cheryauka A., Brehm, S. and W. Christensen, 2006, “Sequential intrinsic and extrinsic geometry calibration in fluoro CT imaging with a mobile C-arm”, *Proceedings of SPIE Medical Imaging*, San Diego. Paper 6141-90.
- [3] Cheryauka A., Barrett, J., Wang, Z., Litvin, A., Hamadeh, and D. Beudet, 2007, 3-D geometry calibration and markerless electromagnetic tracking with a mobile C-arm, *Proceedings of SPIE Medical Imaging*, San Diego. Paper 6509-78.
- [4] Churchill M., Pope G., Penman J., Riabkov D., Xue X., and A. Cheryauka, 2007, Hardware-accelerated cone-beam reconstruction on a mobile C-arm, *Proceedings of SPIE Medical Imaging*, San Diego. Paper 6510-211.
- [5] Feldkamp, L.A., Davis L.C., and J.W.Kress, 1984, “Practical cone-beam algorithm”, *J. Opt. Soc. of America*, 1(6), 612-619.
- [6] GE Healthcare mobile C-arm products for surgical applications, 2006, Available: <http://www.gehealthcare.com/usen/xr/surgery/products/oc9900elite.html>
- [7] GPU Gems 2: Programming Techniques for High-Performance Graphics and General Purpose Computation, 2005, Ed. Matt Pharr, Addison-Wesley, 814 p.
- [8] General-Purpose Computation Using Graphics Hardware (GPGPU). Available: www.gpgpu.org
- [9] Kachelrieß M., Knaup M., amd O. Bockenbach, 2006, “Hyperfast perspective cone-beam backprojection”, *Proceedings of IEEE Medical Imaging conference*, San Diego.
- [10] Li J., Shekhar L. R., and C. Papachristou, 2004, “A ‘brick’ caching scheme for 3D medical imaging”, *Proceedings of IEEE Biomedical Imaging on Micro to Nano*, v.1, pp.563-566.
- [11] Mercury Computer Systems. Available: <http://www.mc.com/industries/lifesciences/medicalimg/>
- [12] Mueller K. and F. Xu, 2006, “Practical considerations for GPU-Accelerated CT”, *IEEE International Symposium on Biomedical Imaging*, Washington D.C.
- [13] Nvidia CUDA: Revolutionary GPU Computing. Available: <http://developer.nvidia.com/object/cuda.html>
- [14] Owens J. D., Luebke D., Govindaraju N., Harris M., Krüger J., Lefohn A. E., and T. J. Purcell, 2007, “A Survey of General-Purpose Computation on Graphics Hardware”, *Computer Graphics Forum*, v. 26, to be published.
- [15] Xu F. and K. Mueller, 2003, “Near-interactive cone-beam computed tomography on commodity PC graphics hardware”, in *Proceedings on IEEE Medical Imaging Conference*, San Diego.
- [16] Xu F. and K. Mueller, 2005, “Accelerating popular tomographic reconstruction algorithms on commodity PC graphics hardware”, *IEEE Transaction of Nuclear Science*.
- [17] Xue X., Cheryauka A., and D. Tubbs. 2006, “Acceleration of fluoro-CT reconstruction for a mobile C-Arm on GPU and FPGA hardware: A simulation study”, *Proceedings of SPIE Medical Imaging*, San Diego. Paper 6142-166.