

White Paper

Current and next-generation GPUs for accelerating CT reconstruction: quality, performance, and tuning

What's inside?

- A performance and quality analysis of GPU accelerated cone-beam CBCT
- A comparison of current and next-generation GPUs

Michael S. Vaz
Imaging Research & Development Engineer
Barco
michael.vaz@barco.com

Matthew McLin
Software engineer
Barco
matt.mclin@barco.com

Alan Ricker
TIG Manager & System Architect
Barco
alan.ricker@barco.com

Michael Vaz, Matthew McLin and Alan Ricker are with Barco Medical Imaging Division, 15425 SW Beaverton Creek Court Beaverton, OR 97006-5168, USA

Barco
Pres. Kennedypark 35
B-8500 Kortrijk, Belgium

BARCO

Visibly yours

ABSTRACT

We present performance and quality analysis for GPU accelerated cone beam computed tomography (CBCT) reconstruction using FDK filtered back projection. We also compare the current generation (2006) Barco MXRT-7100 GPU and a next generation (2007) GPU prototype. Our MXRT implementation is able to reconstruct a 512 x 512 x 340 volume from 625 projections, each sized 1024 x 768, in 42 seconds. This is already very fast, but the next generation prototype GPU is able to reconstruct the same volume in 7 seconds. The presented results also illustrate that while GPUs have immense computational power there is opportunity to tune an implementation to specific hardware, quality requirements and even to the image acquisition geometry.

Index Terms - GPU, CBCT, CT reconstruction, computed tomography, FDK

Copyright © 2007
BARCO n.v., Kortrijk, Belgium

All rights reserved. No part of this publication may be reproduced in any form or by any means without written permission from Barco.

TABLE OF CONTENTS

1. Introduction	4
2. Methods	4
2.1 GPU platforms and implementations.....	4
2.2 Test sets	5
2.3 Quality assessment.....	5
2.4 Features.....	5
3. Results and analysis	6
3.1 Hardware and reconstruction samples	6
3.2 2006 generation GPU performance.....	6
3.3. 2007 generation GPU performance.....	8
3.4. Quality Analysis.....	9
4. Relating to other results in the literature	10
4.1. Comparing to other implementations and platforms.....	10
5. Conclusion and discussion	12
6. Acknowledgement	12
7. References	13

1. INTRODUCTION

CT reconstruction is extremely computationally demanding, which makes hardware acceleration desirable. The use of GPUs for non-graphics purposes, also known as general purpose GPU (GPGPU), has recently become popular due to their massive processing power. While the use of commodity GPUs for CT reconstruction has been proposed and discussed in the literature [1, 2, 3], the quality of GPU accelerated CT reconstruction is not sufficiently addressed.

We have implemented the FDK filtered back projection algorithm for CBCT where data is acquired using planar detectors [4, 5]. We assess the performance and quality for the 2006 generation Barco MXRT-7100 GPU board and also for a 2007 generation prototype GPU board. We evaluate different implementations that are tuned for each board.

While speed is an important factor of CT reconstruction, the more poignant question is whether the implementation is real-time: whether reconstruction is performed at a rate that is equal to or greater than the rate of raw data acquisition (high throughput) and whether the implementation has low latency. A clinically viable implementation would be flexible and asynchronous - it should be able to reconstruct using streamed projections without any constraint on the order in which these projections are received and return the final result quickly after the last projection is acquired. Our real-time streaming CT reconstruction platform does indeed satisfy all of the above criteria and also includes many other desirable features. It was unveiled at RSNA 2006 (<http://www.rsna.org/>) and formally introduced in [6].

In this paper we will demonstrate that there is an opportunity to tune an implementation to a specific GPU and to optimize reconstruction performance for a particular scanning/acquisition geometry. Our analysis provides a thorough quality assessment of our GPU reconstruction. We also compare our performance to the latest published results [1, 2, 3, 7].

2. METHODS

2.1 GPU platforms and implementations

All of our FDK filtered back projection implementations for CBCT discussed in this paper use bilinear interpolation (BL). The overall implementation follows a "real-time streaming" design philosophy and therefore has low latency with high throughput. The algorithm is flexible, asynchronous, and scalable [6]. For the 2006 generation MXRT-7100 board, we have three different implementations: P1, P2, and P3. These implementations differ primarily in their memory access patterns. We also compare two implementations for a 2007 generation prototype GPU board which we identify as Q1, and Q2. In Q2 we make a controlled tradeoff of some numerical precision for significant speed. All other implementations maintain 32-bit floating point precision (FP32) end-to-end.

<i>Test set</i>	<i>Projection size</i>	<i># projections</i>	<i>Recon. Volume size</i>
1	507 x 379	625	512 x 512 x 340
2	1024 x 768	625	512 x 512 x 340
3	800 x 672	1000	512 x 512 x 512
4	1024 x 1024	330	512 x 512 x 512

Table 1: Projection and reconstruction dimensions

2.2 Test sets

For the considered GPUs and implementations, we compare performance for four datasets (table I). All datasets assessed are of actual scanned data.

2.3 Quality assessment

To assess the quality of the GPU reconstruction, we use a floating point C++ implementation as reference. Measures such as maximum absolute error (MaxAE) and error density are used. We also compare difference images (error maps) between the GPU reconstruction and a reference to assess if the error might be correlated with image content and structure. A detailed quality assessment for test set 1 is also presented.

2.4 Features

A useful property of GPU acceleration is that in most cases we can render interim results without significant cost. This is because the required data already resides in graphics memory. As shown in figure 1, our reconstruction platform allows the user to interactively view the reconstruction process live.

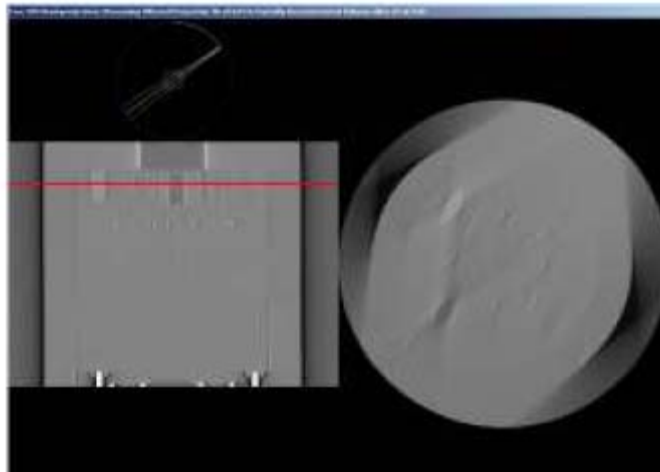


Fig. 1: interactive live reconstruction snapshot for test set 1. The user may “peek” into the GPU to view interim results. The current (filtered) projection that is being back projected into the reconstruction volume is shown on the left hand side window with a reference graphic on top indicating its corresponding angular position. The red line across the projection image indicates the position of the reconstruction slice that is shown on the right hand side window. The user may interactively view different slices during reconstruction by clicking on the current projection and moving the red indicator.

3. RESULTS AND ANALYSIS

3.1 Hardware and reconstruction samples

The GPU implementations were tested using an HP XW 9300 workstation computer with two 2.39 GHz AMD Opteron™ 250 CPUs and 2 GB of RAM. The 1GB MXRT-7100 GPU PCIe board has full 32-bit floating point computation pipeline. Figure 2 shows a volume rendering of our DICOM output for test set 1 (see table I), using our commercially available Voxar 3D software package.

3.2 2006 generation GPU performance

We present results for three different implementations on the 2006 generation MXRT-7100 GPU (P1-3) and two implementations on the 2007 generation GPU (Q1-2). This facilitates intra- and inter-GPU comparisons. Figure 3 shows the performance for the four test sets described in table I across P1-3. Note that no single implementation gives the best performance for all test sets.

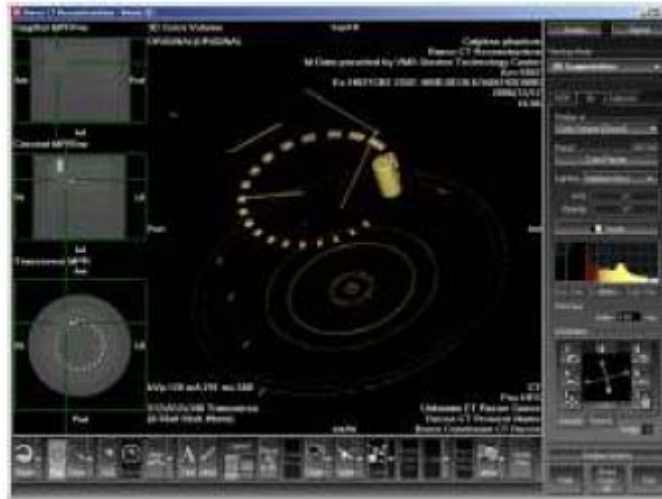


Fig. 2: Reconstructed results are saved in DICOM format and then viewed using Barco's Voxar 3D software. A volume rendering of reconstruction results of test set 1 (Catphan[®] phantom, <http://www.phantomlab.com>) is shown. Axial slice 105 is shown in the lower left sub-window. It corresponds to the high-resolution module of the Catphan[®] phantom which contains a radial gauge which has 1 to 21 line pairs per cm. The concentric rings visible towards the bottom of the VR do not correspond to any structure in the phantom; they are caused by several dead pixels in a single row of the detector panel. This artifact is largely limited to slice 278.

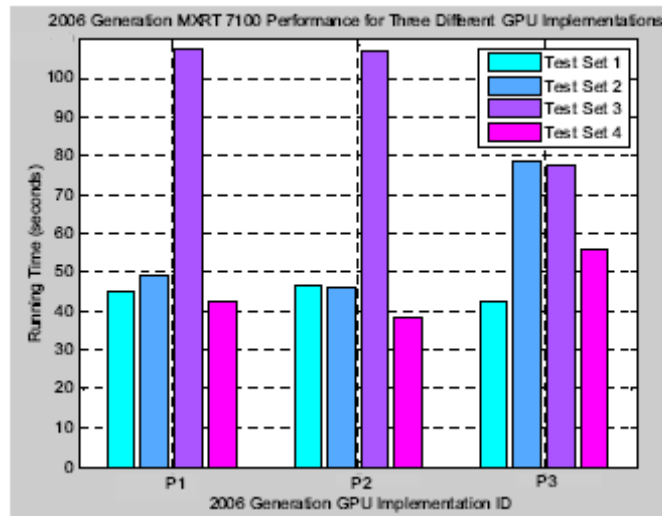


Fig. 3: Performance for the four test sets (table I) on the 2006 generation MXRT-7100 graphics board. Three different GPU implementations, P1-3, produce numerically equivalent reconstruction results. Test sets 1 and 3 have best reconstruction performance using implementation MXRT P3, while the MXRT P2 implementation gives the best performance cases 2 and 4. P1 facilitates interactive visualization.

3.3. 2007 generation GPU performance

Figure 4 depicts the performance for the same four test sets for implementations Q1-2 on a 2007 generation GPU prototype board. We see an average 3-5x performance increase compared to the MXRT (2006). In Q2, we made a controlled trade-off of some numerical precision for a significant gain in speed. Reconstruction quality is discussed further below.

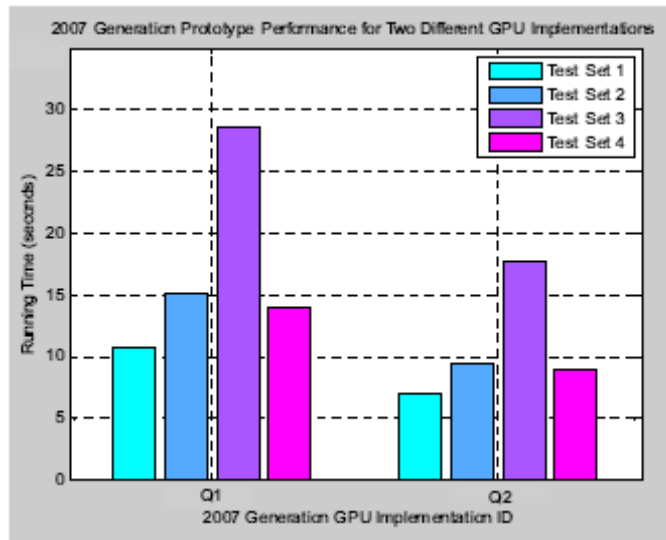


Fig. 4: performance for the four test cases (table I) using the 2007 generation prototype GPU board. Implementations Q1 and Q2, in that order, are on average 3x, and 5x faster than the fastest 2006 generation MXRT performance.

The timing indicated in figure 3 and 4 is measured for the entire reconstruction process which includes the time to read back the reconstructed volume from the GPU to CPU RAM.

3.4. Quality Analysis

We consider the GPU accelerated reconstruction results of test set 1 and seek answers for the following questions:

- What proportion of entire reconstructed volume does not match the C++ reference (i.e. error density)?
- What is the maximum absolute error (MaxAE)?
- Is there any structure in the error and/or any indication that the error might correspond to the image content?

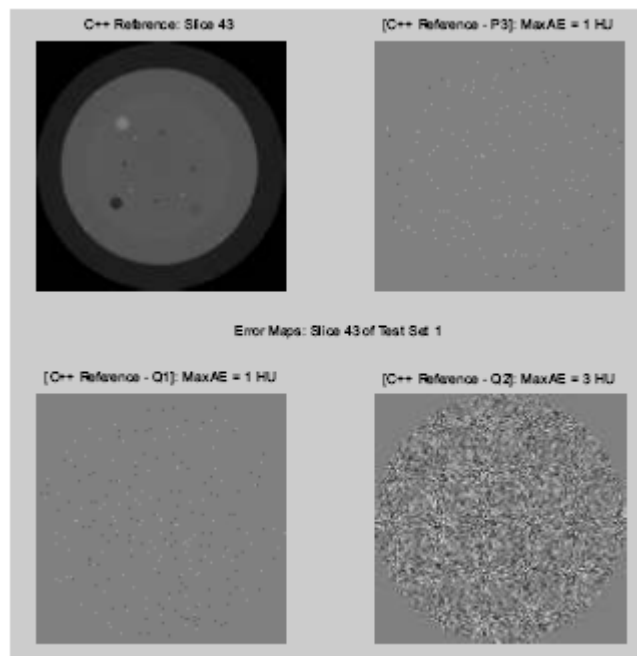


Fig. 5: error maps for different GPU implementations, with respect to the reference C++ result for reconstructed slice 43 of test set 1. The error does not have any apparent structure. Slice 43 corresponds to the slice geometry and sensitometry module of the Catphan[®] phantom.

The error density was 0.5% for P1-3 and Q1. It was 28% for Q2. Note that the quoted error voxel percentages are with respect to cylindrical reconstruction FOV, whose circular cross section occupies 204,265 voxels of each 512^2 slice. A common mistake is to normalize by 512^2 instead, which artificially/falsely reduces the error density. The spatial distribution of differences between the various GPU implementations and the reference C++ reconstruction for Slice 43 of test set 1 are shown in figure 5. Clearly the error density is greater for Q2 than for Q1, but there is no obvious structure present in the shown error maps.

Figure 6 shows that the MaxAE increases in conjunction with the error density increase in Q2. For slice 273 of the analyzed reconstruction volume, the error maps do contain some structure; this particular slice is heavily influenced by acquisition error due to several dead pixels in a single row of the detector plate (figure 2). Thus we consider the higher MaxAE values for slices proximal to slice 278 to be outliers.

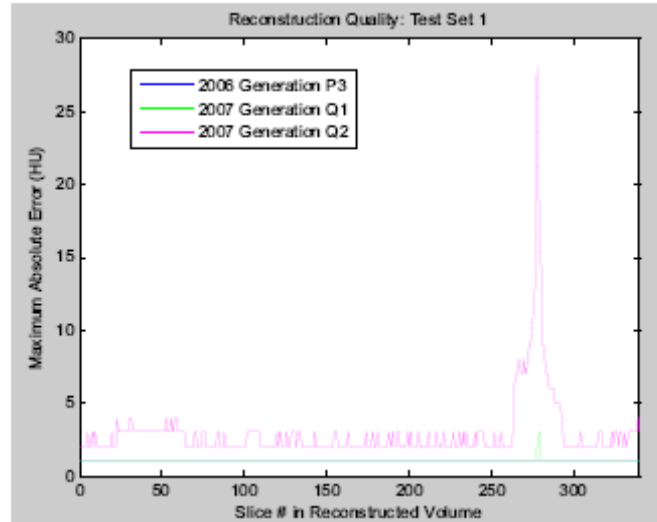


Fig. 6: maximum Absolute Error (MaxAE) for each slice of the 512 x 512 x 340 reconstruction volume of test case 1. P3 has MaxAE of 1 HU for all slices. MaxAE for Q1 is 1 for all slices except slice 278. Q2 has MaxAE of 2-4 HU for all slices except those close to slice 278 which is artifact prone due to an acquisition defect (Fig. 2).

4. RELATING TO OTHER RESULTS IN THE LITERATURE

4.1. Comparing to other implementations and platforms

There have been many efforts to speed up CT reconstruction using CPU, FPGA, CBE (CELL Broadband Engine™) and of course GPUs. While it may be tempting to identify the “best” platform, we have learned that the particular application and the particular implementation are key factors in determining which platform will be favorable [1]. An excellent effort to facilitate comparison of many backprojection (BP) implementations across multiple platforms is made in table III of [7]. Since then [1, 2, 3] have extended it with their latest results. Table II below is our extension of this same table of BP performance. As such, it doesn’t account GPU to CPU read-back time. Our scaled BP times increase across test sets 1, 3, 2, 4 (Table I, Fig. 3 & 4), in that order, indicating that projection size may indeed affect reconstruction timing – most likely caused by caching efficiency issues.

4.2. Danger of using consumer grade GPUs for medical imaging

The high performance results quoted in [1, 2, 3] are all obtained using the Nvidia® GeForce 8800 GTX, which is one of the fastest consumer-level GPU boards available today. In [2] they use that GPU’s hardware-based bilinear (HW-BL) interpolation. Our experiments indicate that the HW-BL on the 8800 GTX is implemented non-separably, where the blending weights have only 8-bit precision. We are consistently able to duplicate the HW-BL output from the 8800 GTX using a limited-precision bilinear interpolator that we implemented in Matlab. This indicates that although this HW-BL does accept 32-bit floating-point (FP32) values as inputs and returns a FP32 value, the result does not have true FP32 bilinear interpolation precision. Rather, it is mathematically equivalent to pre-interpolating by a factor of 256 using FP32 BL and then using nearest neighbor (NN). The “hybrid” method referred to in [7] pre-interpolates by a factor of 2 and then uses NN. Given that the scaled performance quoted in [1, 3] for their fast AG-GPU configuration is approximately equal that of [2] it is possible that similar optimizations are used. We were unable to find any documentation on the 8800 GTX that discussed the 8-bit blending weights for FP32 HW-BL and as such it is possible that many users are unaware. The HW-BL issue is but one instance of where numerical precision may be inadvertently compromised. In our experience, it is necessary to run specifically designed tests to assure end-to-end FP32 precision. We don’t base our solutions on consumer-level GPU boards/drivers. Instead, we partner directly with a GPU manufacturer to develop a medically optimized GPU solution.

This strategic position allows us to assure the integrity and longevity of our GPU-accelerated solution. The Q1 implementation uses full FP32 precision end-to-end as demonstrated in figure 5 and 6. In the table below “direct” refers to full FP32 implementations and as such we use “limited” to refer to the reduced precision of Q2. We support Q2 since there may be applications, such as interventional radiology, where faster processing is preferred over full FP32. For such situations, the HW-BL implementations [1, 2, 3] might suffice as well.

	<i>Type</i>	<i>HW</i>	<i>Time</i>	<i>Comment</i>
Kachelreiss et al. [7]	LI/F32	CBE	27.2 s	Direct
	LI/F32	CBE	13.6 s	Hybrid
Churchill et al. [2]	LI/F32	GPU	12.8 s	Direct (limited?)
Müller et al. [1,3]	LI/F32	GPU	12.7 s	Direct (limited?)
Vaz et al. [this]	LI/F32	GPU	11.9 – 19.0 s	Q1 – direct
	LI/F32	GPU	7.2 – 11.3 s	Q2 - limited

Table 2: extension of table III in [7]: backprojection performance scaled to 512 projections and 512³ volume

5. CONCLUSION AND DISCUSSION

We have presented a detailed performance and quality assessment for our CT reconstruction platform, where FDK CT reconstruction is accelerated using the 2006 generation Barco MXRT-7100 GPU or our 2007 generation GPU prototype. It is believed that GPUs are improving at triple Moore's law [1] and our work confirms this. The commodity nature of GPUs keeps their cost low, thereby making GPUs an attractive acceleration platform for CT reconstruction. The fact that we can accurately reconstruct a 512^3 volume from 1000 projections in less than 20 seconds using a graphics card proves the awesome computational power of GPUs.

The presented results demonstrate that there is an opportunity to optimize implementations considering specific GPU architecture features, acquisition geometry and required numerical precision. We also demonstrated that we can successfully maintain full 32-bit precision end-to-end at real-time speeds.

Our real-time streaming architecture is designed such that the fully reconstructed volume is available soon after the last projection is acquired [6]. We perform pre-weighting and filtering on the CPU in parallel to backprojection on the GPU.

This partitioning takes best advantage of each processor to give the highest throughput. Our performance scales with multiple GPUs [6] and the live reconstruction visualization feature can be useful for catching and correcting acquisition problems as they occur. We believe our "GPU+CPU" solution will continue to offer the best price per performance and image quality over time.

6. ACKNOWLEDGEMENT

Test sets 1 & 2 were provided by Varian Medical Systems, Ginzton Technology Center.

7. REFERENCES

- 1 K. Mueller, F. Xu, N. Neophytou, "Why do commodity graphics hardware boards (GPUs) work so well for accelerating computed tomography?" in *Proc. SPIE Medical Imaging*, 2007.
- 2 M. Churchill, G. Pope, J. Penman, D. Riabkov, X. Xue, A. Cheryauka, "Hardware-accelerated cone-beam reconstruction on a mobile C-arm," in *Proc. SPIE Medical Imaging*, 2007.
- 3 F. Xu, K. Mueller, "Real-time 3D computed tomographic reconstruction using commodity graphics hardware," *Phys. Med. Biol.*, Vol. 52, No. 12, 3405-3419, 2007.
- 4 H. Turbell, "Cone-beam reconstruction using filtered backprojection," *Doctoral Thesis, Dept. of Electrical Engineering, Linkoping University, Sweden*, 2001.
- 5 J. Hsieh, *Computed Tomography Principles, Design, Artifacts, and Recent Advances*, SPIE Press, 2003.
- 6 M. S. Vaz, Y. Sneyders, M. McLin, A. Ricker, T. Kimpe, "GPU accelerated CT reconstruction for clinical use: quality driven performance," in *Proc. SPIE Medical Imaging*, 2007.
- 7 M. Kachelrei, M. Knapp, O. Bockenbach, "Hyperfast perspective conebeam backprojection," in *Proc. IEEE Nuclear Science Symposium*, Vol. 3, 1679-1683, 2006.