2014 IEEE 22nd Annual International Symposium on Field-Programmable Custom Computing Machines

FPGA Acceleration for Simultaneous Medical Image Reconstruction and Segmentation

Peng Li[1], Thomas Page[3,2], Guojie Luo[2,4], Wentai Zhang[2], Pei Wang[2], Peng Zhang[1],
Peter Maass[3], Ming Jiang[2,4] and Jason Cong[1,2,4*]
[1]University of California, Los Angeles, [2]Peking University, [3]University of Bremen, [4]UCLA/PKU Joint Research Institution

I. Introduction

The conventional approach of computed tomography (CT) is to solve each image processing task individually in sequence: 1) image reconstruction; 2) post-processing; 3) segmentation. An obvious drawback is that the measured data is only used once at the first step, and the possible errors, from noises in the measured data, inappropriate modeling, or inappropriate parameters, are not easy to be corrected and will be propagated into the later steps. As a consequence, approaches that combine the reconstruction and the specific processing task have become popular [1], [2]. In this work, we adopt an iterative algorithm with simultaneous reconstruction and segmentation using the Mumford-Shah model[3], which can be applied not only to regularize the ill-posedness of the tomographic reconstruction problem, but also to compute segmentation directly from the measured data. The Mumford-Shah model is both mathematically and computationally difficult. In this paper, we accelerated this computation and data intensive application by FPGA devices and achieved 9.24X speedup over the conventional CPU implementation.

II. Design Methodology and Optimizations

In this paper, we use high-level synthesis (HLS) flow to design and implement the FPGA accelerator. The concept-proofing Matlab programs are first manually translated into an HLS friendly C programs, and then optimized from algorithm-level, inner-module and inter-module for efficiency.

Profiling of the algorithm execution shows that forward/backward projections dominate the total execution time by 97%. The forward projection maps a function f into the set of its line integrals, while backward projection maps the line integrals back into an image. The first optimization technique we develop is to reduce the number of projections using the linear property of the forward/backward projections to transform some projections to the linear combination of previously projected images. With the algorithm-level optimization, the number of projections in one iteration are reduced from four to two.

The second optimization is to parallelize the computation kernels: forward/backward projections. The images are partitioned into disjoint tiles and projected separately before the final accumulation. The boundary calculation for image tiles with a fixed x-ray beam is complex and resource consuming. Therefore, we precompute the results and save them in lookup tables. Two major factors are considered in selecting degrees of parallelism: resource limitation and communication overhead. The problem can be formulated as a posynomial optimization problem and can be solved by geometric programing. For our target application and platform, the optimal degrees of parallelism for forward/backward projections are 22 and 16 respectively.

The last optimization is inter-module optimizations including common expression elimination, loop merging and data streaming. With inter-module optimization, 3 loops and 2 temporal arrays can be eliminated.

III. Experimental Results

Xilinx Virtex-7 board VC707 is selected to be the target hardware platform in our experiment. Xilinx Vivado Design Suite 2013.1 is invoked by our automated system level design tool, which generates the auxiliary modules automatically. The Shepp-Logan phantom [4] is used as the test input with an image size of 512*512. The alternate iteration counts and minimize image/edge iteration counts are all set to 10. Therefore, the reconstruction and edge indicator are updated 100 times each in the entire process. We have implemented several FPGA designs and also CPU/GPU versions for comparison. Table I shows the execution time, power and energy consumption of various implementations. From the table, we can see that the speedup of the optimized FPGA implementation over a reference CPU implementation is 9.24X. The energy efficiency of the optimized FPGA design is 168.9X over the CPU implementation and 19.2X over the GPU implementation.

TABLE I: Experimental Results of Various Implementations

Implementations	Exe. Time(s)	Speed up	Power (W)	Energy (kJ)	Energy Eff.
CPU (Xeon E5-2430)	453	1	95	43.0	1
GPU (Radeon HD 7850)	17	26.65	289	49.0	8.8
FPGA-Baseline	441	1.03	4.8	2.1	20.3
Algorithm Optimization	221	2.05	4.8	1.1	40.6
Inter Module Optimization	218	2.08	4.8	1.0	41.1
Parallel Kernels	49	9.24	5.2	0.25	168.9

IV. Acknowledgment

This work was supported partly by C-FAR, one of six centers of STARnet, a Semiconductor Research Corporation program sponsored by MARCO and DARPA, the National Basic Research Program of China (973 Program) (2011CB809105), the National Science Foundation of China (61121002, 61202073), and MoE/RFDP 20120001120124.

References

[1] R. Ramlau, E. Klann, and W. Ring, "Simultaneous reconstruction and segmentation for tomography data," *PAMM*, vol. 7, no. 1, pp. 1 050 303–1 050 305, 2007.

[2] Q. Zhang, R. Plemmons, D. Kittle, D. Brady, and S. Prasad, "Joint segmentation and reconstruction of hyperspectral data with compressed measurements," *Appl. Opt.*, vol. 50, no. 22, pp. 4417–4435, Aug 2011.

[3] M. Jiang, P. Maass, and T. Page, "Regularizing properties of the mumford-shah functional for imaging applications," *Inverse Problems, in press*, 2014.

[4] A. C. Kak and M. Slaney, *Principles of Computerized Tomographic Imaging*. IEEE Press, 1998, available online at http://www.slaney.org/pct/pct-toc.html.

[*] Prof. Jason Cong is a distinguished visiting professor at Peking University.

978-1-4799-5111-6/14 $31.00 © 2014 IEEE
DOI 10.1109/FCCM.2014.54