Implementation of Exact Reconstruction Algorithm of Helical Cone-Beam CT on Cell Broadband Engine Architecture

Jin Yannan He Qing Han Zhenxing Cao Liji

July 5, 2007

Introduction

Helical cone-beam ct is an important medical imaging modality for its high longitudinal resolution and efficient use of radiation dose. The data acquisition system consists of an x-ray source and a planar detector attached to a rotating gantry while the patient is translated steadily through the gantry. Therefore, from the perspective of the patient, the trajectory of the x-ray source forms a helix. The cone-beam x-rays generated by x-ray source pass through the patient and the attenuation signals are recorded by the planar detector on the other side of the patient. The task of ct reconstruction is to recover the 3d images from the projection data.

Katsevich algorithm[1], proposed in 2002, was the first exact reconstruction algorithm for helical ct. It is a major breakthrough in the development of ct reconstruction algorithm. However, the intensive computation required by this algorithm prohibits its clinical use, especially when the amount of projection data becomes increasingly large. Parallel computing can be an effective approach to resolving the problem of heavy computation burden. Cell broadband engine architecture (cbea), a novel computing architecture, is a very promising mode for this purpose. The cell processor offers an immense computing power due to its high level of parallelism, its high clock rate and its enormous internal communication bandwidth. Moreover, it can be programmed using high level programming languages.

In this work, we implement the exact bpf (backprojection-filtration) reconstruction algorithm[2] on a cell/ps3 and perform numerical simulations. We parallelize and optimize the exact bpf reconstruction algorithm to accelerate the imaging process. The simulation results demonstrate a significant reduction of reconstruction time and indicate the potential of cbea's application in medical image reconstruction.

Exact reconstruction algorithm

The trajectory of helix can be mathematically written as

$$\boldsymbol{a}(t) = \left\{R\cos(t), R\sin(t), \frac{h}{2\pi}t\right\} \tag{1}$$

where R is the helix radius and h is the helix pitch. For an object which is compactly supported in the cylindrical region $\mathbf{U} = \{(x,y,z)|x^2+y^2 \le r^2\}$, $0 < r < R$, where r is the radius of the object cylinder. The BPF reconstruction formula is given as follows:

$$f(\boldsymbol{x}) = \frac{1}{2\pi}\mathcal{H}_{\boldsymbol{a}_2-\boldsymbol{a}_1} \circ \mathcal{D}^{\#} \circ \partial \circ \mathcal{D}f(\boldsymbol{x}) \tag{2}$$

where $\mathcal{D}f$ is the cone-beam transform defined as

$$\mathcal{D}f\big(\boldsymbol{a}(t), \boldsymbol{\alpha}_{\boldsymbol{x}}(t)\big) = \int_0^\infty f\big(\boldsymbol{a}(t) + s \cdot \boldsymbol{\alpha}_{\boldsymbol{x}}(t)\big)\mathrm{d}s \tag{3}$$

The operator ∂ is the derivative in respect to the variable t of $\mathcal{D}f\,(\boldsymbol{a}(t), \boldsymbol{\theta})$, and $\mathcal{D}^{\#}f$ is the backprojection operator

$$\mathcal{D}^{\#}g(\boldsymbol{x}) = \int_{I_x} g\big(\boldsymbol{a}(t), \boldsymbol{\alpha}_{\boldsymbol{x}}(t)\big)\frac{\mathrm{d}t}{\|\boldsymbol{x} - \boldsymbol{a}(t)\|} \tag{4}$$

where $I_{\boldsymbol{x}} = [t_1, t_2] \in \mathbb{R}$ referred to as a chord interval passing through $\boldsymbol{x} \in \Omega$, $\|\boldsymbol{x} - \boldsymbol{a}(t)\|$ is the Euclidean norm of the vector $\boldsymbol{x} - \boldsymbol{a}(t) \in \mathbb{R}^3$, and the unit vector

$$\boldsymbol{\alpha}_{\boldsymbol{x}}(t) = \frac{\boldsymbol{x} - \boldsymbol{a}(t)}{\|\boldsymbol{x} - \boldsymbol{a}(t)\|} \in S^2 \tag{5}$$

points to $\boldsymbol{x}$ from the source location $\boldsymbol{a}(t)$.

$\mathcal{H}_{\boldsymbol{e}}$ is the Hilbert transform along a given direction $\boldsymbol{e} \in \mathbb{R}^3$, which can be defined via Fourier transform as

$$\mathcal{H}_{\boldsymbol{e}}\varphi(\boldsymbol{x}) = \frac{1}{(2\pi)^3\mathrm{i}}\int_{\mathbb{R}^3} \mathrm{sgn}(\boldsymbol{e} \cdot \boldsymbol{v})\hat{\varphi}(\boldsymbol{v})e^{\mathrm{i}\boldsymbol{x}\cdot\boldsymbol{v}}\mathrm{d}\boldsymbol{v} \tag{6}$$

where $\hat{\varphi}(\boldsymbol{v})$ is the inverse Fourier transform of the function $\varphi(\boldsymbol{x})$.

The backprojection-filtration formula described above can be implemented in the following steps:

1. For each projection, calculate the differentiation of each data with respect to rotation angle t, using the method described by Yu *et al*[4].
2. Calculate the backprojection $\mathcal{D}^{\#}g(\boldsymbol{x})$ along the π-line $\boldsymbol{a}_2 - \boldsymbol{a}_1$, using equation (4) and (5).
3. Reconstruct the object function $f_\pi\left(x_\pi, t^{\min}, t^{\max}\right)$ along π-lines by performing inverse Hilbert transform using the method described in [5]. Here $t^{\min}$ and $t^{\max}$ are the rotation angle corresponding to the start and end point of the π-line.
4. Rebin the reconstructed result $f_\pi\left(x_\pi, t^{\min}, t^{\max}\right)$ into the Cartesian coordinate system by determining the π-line for each grid in Cartesian coordinate system.

CBEA BASED OPTIMIZATION

Decomposing the computing task

The BPF exact CT algorithm can be very time consuming as the data set grows. This restricts the clinical use of this algorithm on a general purpose, low cost computer. However, the computation of slices is fully independent, which makes it ideal for parallel programming and CELL architecture.

Considering the conventional thread model, we put the computing task of every slice in a dedicated thread with input and output. The input data of the thread is the differentiation of the recorded sensor data, which are read-only and not modified after the global initialization is finished. The output data between different slices are also independent and stored separately. The characteristic of input and output implies that the threads have no interdependence and race.

These dependence and lock free threads perfectly fit to SPE of the PS3. Moreover, proper optimization on vectorization and DMA arrangement can further increase the performance.

For optimal utilization of SPE resources, a lightweight user space scheduler runs in the main thread in PPE context. It simply makes the SPE threads process a new slice as soon as the previous one is completed without any waiting. This also saves the time for extra image reloading and other SPE initialization stuff. The communication between SPE's and PPE are facilitated by the mailbox mechanism provided by the CBEA.

Vectorization

The process of backprojection mainly consists several loops, due to the integral expression of the algorithm. This gives the algorithm the potential to utilize powerful SPU vectorization capabilities.

The most inner loop of our implementation is against different angles. The process is composed of many floating point calculations and is independent with each other, which is a good point getting vectorized. The large amount of intrinsics available in SPU C language extension are widely used to achieve better performance. Because the boundary of the loop variable is not 4 aligned, we use a wider-boundary-and-mask method to avoid possible branching instructions in the object code.

Another good vectorization candidate is the interpolation routine used in backprojection. This routine uses a two dimensional linear interpolation to estimate more accurate data. This two dimensional linear interpolation needs a 4 point calculation which also fits the vector processing very well.

DMA control

MEMORY FLOW CONTROLLER (MFC), provides the DMA mechanism for the data transfer between PPU and SPU. Although it's much faster than conventional device DMA, it is still relatively slow with respect to SPU local storage. Thus, it is often the performance bottleneck of the computing task running on SPU. In our implementation, it is also proved that a sophisticated design of DMA operation is the key for good performance.

Because the common local storage size of an SPE is only 256k bytes, it's neither able nor suitable to copy all the data from PPU to SPU at one time. Our approach is to use DMA-on-demand operations.

The backprojection algorithm consists of a large amount of get DMA (read from the perspective of SPUs) and a much smaller amount of put DMA. The implementation uses a multi-buffered paradigm to minimize the latency caused by the DMA. The implementation uses a lazy blocking sytle of DMA access, the get and put commands are issued as early as possible while the completion is checked as late as possible. For better performance, we rewrite the backprojection loop into a "pipelined" like structure: in a single loop cycle, the DMA commands are issued right after the necessary parameters are calculated, and in the rest of the loop cycle, the results of the last DMA group is used to calculate the final result. This gives the system a very good performance gain.

In addition, we carefully choose the data structure in the main memory to minimize the DMA count. As mentioned above, the 4 point interpolation is composed of 4 data points, which needs 4 DMA s. Our pre-adjusted data structure made them adjacent in 2 groups, which reduces the DMA count to 2 and vastly increases the speed.

Numerical experiments

To validate and evaluate the parallel reconstruction scheme on CBEA proposed above, we perform a preliminary simulation on a SONY PS3 console using the Shepp-Logan phantom. The CELL processor on PS3 runs at 3.2GHZ, with one PPE and 7 SPE's (only 6 can be accessed under Linux environment). Memory size of PS3 is 256MB which is a liitle small for this algorithm.

The reconstruction result is a $256 \times 256 \times 256$ matrix, each element corresponding to the gray value of a point. We use the gray scale window [0.99, 1.05] to make low contrast features visible. As shown in the attachment, the reconstructed images agree well with the original Shepp-Logan phantom.

The benchmarks of a parallel algorithm are quantified in terms of speedup S_p and parallel efficiency η, which are respectively defined as

$$S_p = T_s / T_{n_p} \quad \text{and} \quad \eta = S_p / n_p$$

where n_p is the number of SPE's, T_s is the total execution time when one SPE is used, T_{n_p} the total parallel execution time when n SPE's are used.

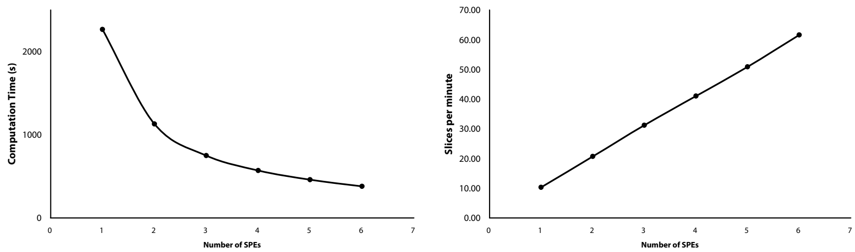

Figure 1: scalability of SPE based BPF implementation

Discussion and conclusion

In this work, we showed a parallelized implementation scheme of the BPF exact reconstruction algorithm on the novel CELL processor. The gray values of the reconstructed images agree well with that of the phantom. In the simulation study, we significantly reduced the reconstruction time from 2.5 hours (on a 2.8GHZ PC) to 10 minutes (on a PS3 equipped with 6 SPE's). The speedup of this architecture is very close to linear, so the reconstruction time can be reduced further by adding the number of SPE's, which will make the exact reconstruction algorithm clinically practical.

It should be noted that the computing architecture based on CELL processor provides a platform which enables us to implement algorithms with a high level programming language. This advantage not only simplifies the development of reconstruction

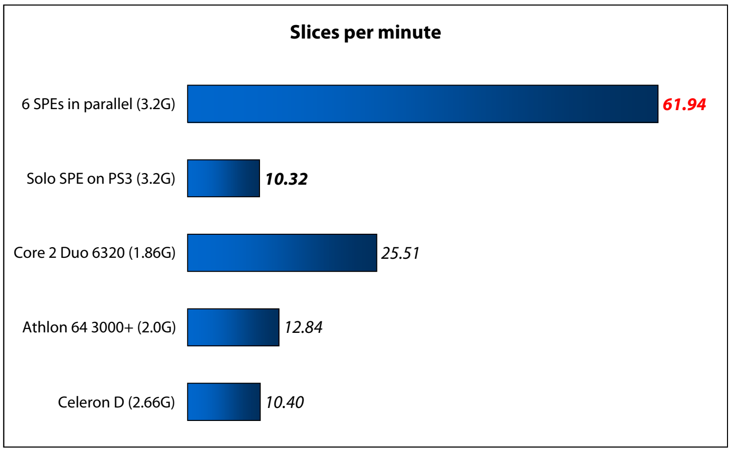

Figure 2: performance comparison

devices, but also makes it very convenient to implement different algorithms on the same device. For instance, we can develop two parallel reconstruction schemes: FDK (Feldkmap-Davis-Kress) algorithm and exact BPF algorithm. The FDK is widely used in commercial CT for it efficiency and it has been reported that CBEA has the potential to conduct real-time reconstruction using FDK algorithm [6]. Therefore, we can produce the real-time preliminary results using FDK algorithm during the CT scanning process; then reconstruct the exact results in a few minutes after the scanning. Both algorithms can be developed and implemented on the same CBEA without increasing the cost of hardware.

In conclusion, the CBEA is a very promising computing architecture for CT reconstruction, especially for the exact reconstruction algorithms which require intensive computing capacity. CBEA may provide an effective and economical solution for the development of commercial image reconstruction device.

References

[1] A. Katsevich, "An improved exact filtered backprojection algorithm for spiral computed tomography", *Adv. Appl. Math. 32, 681-697, 2004*

[2] Y. Zou and X. Pan, "Exact image reconstruction on PI-lines from minimum data in helical cone-beam CT", *Phys. Med. Biol. 49, 941-959, 2004*

[3] S. Zhao, H. Yu and G. Wang, "A unified framework for exact cone-beam reconstruction formulas", *Med. Phys., 32, 1712-1721, 2005*

[4] H. Yu and G. Wang, "Studies on Implementation of the Katsevich Algorithm for Spiral Cone-beam CT", *Journal of X-ray Science and Technology; 12(2), 2004*

[5] F. Noo, R. Clackdoyle and J. D. Pack, "A two-step Hilbert transform method for 2D image reconstruction", *Phys. Med. Biol. 49, 3903-3923, 2004*

[6] H. Scherl, M. Koerner, H. Hofmann, W. Eckert, M. Kowarschik, J. Hornegger, "Implementation of the FDK algorithm for cone-beam CT on the cell broadband engine architecture", *Proc. of SPIE, vol. 6510, 651058-1, 2007*