High Performance Model Based Image Reconstruction

Xiao Wang
School of Electrical
Engineering, Purdue
University
wang1698@purdue.edu

Charles Bouman
School of Electrical
Engineering, Purdue
University
bouman@purdue.edu

Samuel Midkiff
School of Electrical
Engineering, Purdue
University
smidkiff@ecn.purdue.edu

ABSTRACT

In Computed Tomography (CT) methods, Model Based Iterative Reconstruction (MBIR) produces higher quality images than commonly used Filtered Backprojection (FBP) but at a very high computational cost. We describe a new MBIR implementation, PSV-ICD, which significantly reduces the computational cost of MBIR while retaining its benefits. It describes a novel organization of the scanner data into *super-voxels* (SV) that, combined with a *super-voxel buffer* (SVB), dramatically increases locality and prefetching while enabling parallelism across SVs. Experimental data is presented showing an average speedup of $187X$ on 20 cores.

1. INTRODUCTION

Modern imaging systems have increasingly used computation to form useful images from raw sensor data. Perhaps the most classic example of a computational imaging system is X-ray tomographic imaging. Among different reconstruction methods, MBIR [2] has been shown to result in higher quality images compared to FBP but with a very high computational cost. In typical applications, iterative reconstruction methods may require a factor of 10 to over 100 times the computation of FBP. To reduce the amount of computations in MBIR, iterative coordinate descent (ICD) [2] is used to speed up numerical convergence. However, currently it is believed that ICD requires operations that are difficult to parallelize [1,2] and, more importantly, the data layout access pattern exhibits *poor* cache locality. ICD requires that 2D arrays of memory be accessed in sinusoidal patterns, shown in Fig. 1(a). Therefore, the computation speed of ICD iterations is limited by the speed of these sinusoidal memory accesses.

2. THE SUPER-VOXEL (SV)

For the cache locality issue, we use the concept of the super-voxel (SV). A SV is a group of voxels that are contiguous in the image space and whose measurement data (from scanners) is likely to be close together in memory. Thus, operat-
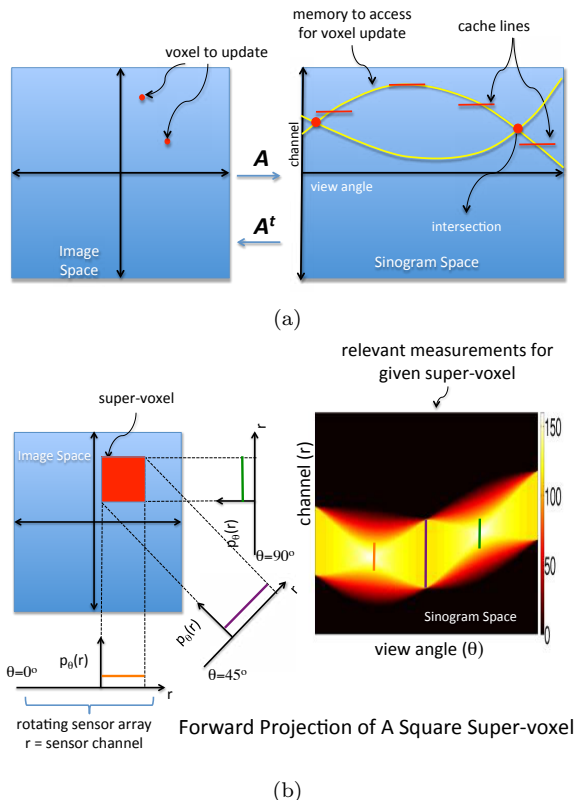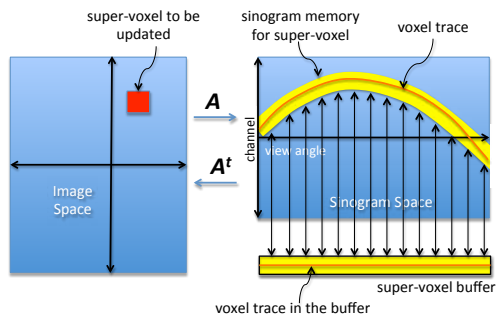
(a)

(b)

Figure 1: (a) Illustrates the forward projection of two voxels. Notice that each single voxel in the image domain traces out a sinusoidal pattern in the sinogram domain. (b) Shows the forward projection of a square SV.

ing on the data for these voxels will increase temporal and spatial locality. This forward transformation from the image space to the sinogram space is shown in Fig. 1(b) where bright points (more yellow and white) in the sinogram space indicate high level of data reuse in the SV.

SVs function by reducing data cache misses. However, SVs do not improve the prefetch rate because most SVs are not centered in the image and their measurement data in the sinogram space follows a sinusoidal band pattern, shown in Fig. 1(b). Additional improvements can be achieved by improving the ability of the hardware to perform prefetching. To improve prefetching, the memory accesses in a SV are copied to the super-voxel buffer (SVB). The SVB lays out

Factor	BS	SV	PSV(4)	PSV(20)
E_F	0.41%	5.95%	4.0%	3.70%
$T_r \pm SE$	253 ± 20	15 ± 0.7	5.6 ± 0.2	1.35 ± 0.1
Speedup		$16.86X$	$45.17X$	$187.40X$

Table 1: Table of performance measurement for all algorithms discussed in this paper. Column 2 is the baseline ICD algorithm. Column 3 is ICD algorithm with SV and SVB design. Columns 4-5 are PSV-ICD with 4 cores and 20 cores respectively. Row 2 ,E_F, is the average GFLOP efficiency. Row 3 ,$T_r \pm SE$, is the average reconstruction time in seconds with the standard error. Row 4 is the speedup of $T_r \pm SE$ at 20 cores comparing with the sequential baseline ICD.

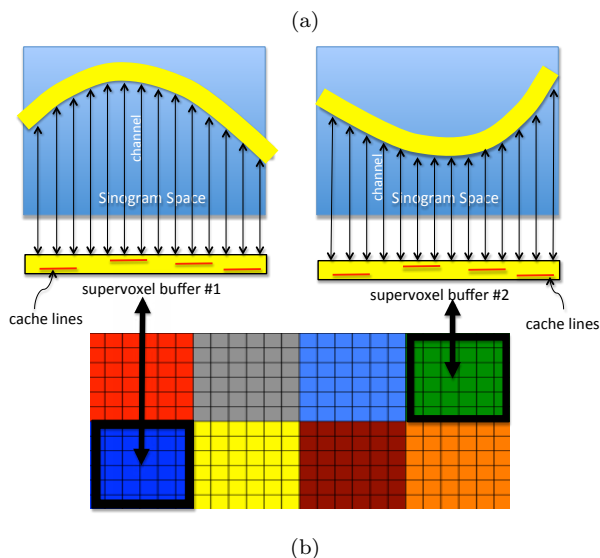Super-Voxel Buffer Management

(a)

(b)

Figure 2: (a) Illustrates this band pattern in which a red block represents the SV to be updated. Sinogram entries are copied to a SVB shown as a straight yellow band. (b) Parallel SVs update with separate augmented SVBs.

the SV data so that memory accesses follow a "straight line" pattern that is ideal for hardware prefetching. Fig. 2(a) graphically illustrates the concept of SVB. Thorough analysis about the impact of SVBs on the performance shows an average tenfold increase of the prefetch hit rate.

3. THE PARALLEL SUPER-VOXELS

In the CT image reconstruction community, the most prevailing parallel voxels update mechanism is called Grouped Coordinate Descent (GCD) [1] with a speedup of 2.3 on 16 cores. Since different voxels' traces have intersections, GCD requires an atomic update at every intersection, increasing lock waiting overhead. To have fewer intersections, GCD simultaneously updates voxels that are widely separated. Although this leads to better parallel performance by reducing the lock waiting overhead, cache locality is worse. Our algorithm, called PSV-ICD, reduces this lock waiting overhead, significantly increases cache locality and also shows good parallel computation performance. Table 1 shows the average reconstruction time by using PSV-ICD on a standard dataset described in the poster. PSV-ICD achieves an

average speedup of $187.40X$ on 20 cores.

In PSV-ICD, multiple SVs that are far apart in the image space update in parallel, similar to GCD, to minimize interference and lock waiting overhead. However within a SV, updates proceed sequentially, as in Sec. 2. In addition, each SV has an individual SVB, denoted by SVB^k for the kth SV. We call this technical innovation the *Augmented SVB*. This augmented SVB, illustrated in Fig. 2(b), entails parallelization and allows each core to access its data without interfering with other cores. The data for multiple SVs is held in cache at the same time, increasing cache pressure, which can be controlled by tuning the size of the SVs.

The difficulty in Inter-SV parallelism is that the SVBs for different SVs overlap in the sinogram, shown in Fig. 2(b). This means the individual SVBs' data must be combined into the full sinogram at the end of SV updates. A simple solution to update atomically for simultaneous SV updates will not work because a SV update will overwrite other SVs' updates. Therefore, we create an extra error buffer, denoted by $SVB^k_{\Delta e}$, for k^{th} SV. $SVB^k_{\Delta e}$ keeps all of the changes of the data relative to the kth SV. When k^{th} SV has been updated, the changes of the data are atomically **added** to the full sinogram from $SVB^k_{\Delta e}$. Intrinsically, this approach reduces lock waiting overhead because all of the local updates are combined into one update in the full sinogram. Voxel updates within a SV can be performed asynchronously while other SVs are being processed, so there is no lock waiting overhead within a SV.

4. REFERENCES

[1] J. Fessler, E. Ficaro, N. Clinthorne, and K. Lange. Grouped-coordinate ascent algorithms for penalized-likelihood transmission image reconstruction. *IEEE Trans. Med. Img.*, 16(2), 1997.

[2] J. Zheng, S.Squib, K.Sauer, and C.Bouman. Parallelizable bayesian tomography algorithms with rapid, guaranteed convergence. *IEEE Trans. Img. Processing*, 9(10), 2000.