Low-Complexity Construction of X-Ray Backprojection Matrix Without Race Condition

Mert Hidayetoğlu[1], Tekin Biçer[2], Doğa Gürsoy[2,3], and Weng Cho Chew[1] Wen-Mei Hwu[1]

[1]*University of Illinois at Urbana-Champaign, Urbana, IL 61801, USA*
[2]*Argonne National Laboratory, Lemont, IL 60439, USA*
[3]*Northwestern University, Evanston, IL 60208, USA*
hidayet2@illinois.edu

***Abstract*— Memory-bound implementation of iterative x-ray image reconstruction requires explicit sparse-matrix representation of the backprojection operator. Naive construction of the backprojection matrix has either a high computational complexity or race conditions, constituting a bottleneck for efficient implementation and parallelization. This paper presents a multilevel algorithm for low-complexity and parallel construction of the x-ray backprojection matrix. Results demonstrate 1,247x parallel speedup on up to 64 Intel Knights Landing nodes.**

I. Introduction

Iterative x-ray reconstruction involves projection and backprojection operations in each iteration, whose memory-bound implementations require explicit sparse matrix representations. The projection matrix can be constructed with a ray-tracing algorithm. A fast ray-tracing has $\mathcal{O}(NM)$ computational complexity, where N is the number of pixels in one dimension of the imaging domain and M is the number of illuminating rays. The backprojection matrix can be trivially constructed by transposing and scaling the projection matrix, however, the sparse matrix transposition suffers from race conditions, and therefore is not parallelizable. Advanced approaches for general sparse matrix transforms are proposed [1], nevertheless, their complexity are not optimal since they do not exploit ray-tracing nature of the problem. Another parallel approach for the matrix construction is to test each pixel with all rays to find out which rays cross a given pixel. The testing can be performed with a line clipping, e.g., Liang-Barsky [2], algorithm, which also gives the intersection length. However, the computational complexity of a naive testing algorithm is $\mathcal{O}(N^2M)$ since each pixel tests all rays. In this paper, we propose a multilevel algorithm for obtaining $\mathcal{O}(NM)$ computational complexity by exploiting the ray-tracing nature of the problem.

II. Multilevel Construction of the Backprojection Matrix

Consider a scanning scenario where the imaging domain is discretized with $N \times N$ pixels and is scanned by M illuminating rays. A naive construction of the backprojection matrix tests all rays for each pixel to find out the crossing rays, yielding $\mathcal{O}(N^2M)$ operations. However, a multilevel testing hierarchically partitions the imaging domain into subdomains, and lets each subdomain tests only those rays which crosses its parent subdomain. The domain partitioning is performed in a top-down fashion until the rays which crosses each pixel-level subdomain is found. As a result, the multilevel testing performs $\mathcal{O}(NM)$ operations by avoiding redundant testings.

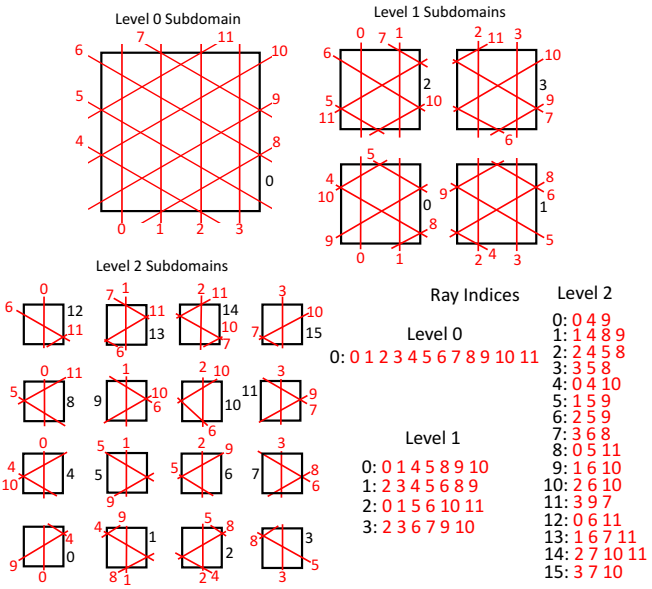

Fig. 1. Multilevel partitioning of the scanning geometry. Children subdomains test rays that crosses their parents only.

Fig. 1 depicts the multilevel construction of the backprojection matrix for a scanning geometry with $N = 4$ and $M = 12$. Initially, the top-level subdomain is considered, where all rays crosses the imaging domain (which is trivial). Then, level-1 subdomains are tested with the rays crossing the top-level subdomain. Secondly, level-2 subdomains (corresponding to pixels) are tested with rays crossing their level-1 parents. For example, pixel 0 tests only rays 0, 1, 4, 5, 8, 9, and 10, instead of testing all rays from 0 to 11.

Fig. 1 also lists ray indices that crosses subdomains at each level. In the pixel level, these the ray indices correspond to the row indices of non-zero elements of the sparse backprojection matrix. The non-zero elements store the length of pixel-ray intersections, which are found in the last step of the matrix construction. Efficient representation of the sparse matrix is achieved with compressed sparse row (CSR) format, where the index and weight data structures are linearized and the starting

index of each row are stored in a displacement array. It is also worth noting that the proposed algorithm preserves the order of rays, which is useful for applications where preserving data locality is important.

In the multilevel testing scheme, number of subdomains quadruples and the number of rays per subdomain decreases approximately by half at each increasing level, and therefore the number of tests doubles. As a result, there are $\mathcal{O}(\log N)$ levels and the total number of tests is $\mathcal{O}(NM)$. On each level, testings are parallelizable over the subdomains, where the multilevel scheme provides good granularity for parallelization, especially at the lower levels with large number of subdomains. The proposed implementation no data duplication or race conditions, and therefore suitable for massively-parallel computing architectures.

III. Scaling Results

For demonstrating performance the proposed multilevel matrix reconstruction algorithm, we perform two set of experiments on Intel Knights Landing (KNL) nodes with 64 cores each. In the first experiment, we show the asymptotic scaling by constructing projection and backprojection matrices for various sizes of problems. In the second experiment, we show the parallel efficiency by performing matrix constructions among various number of cores.

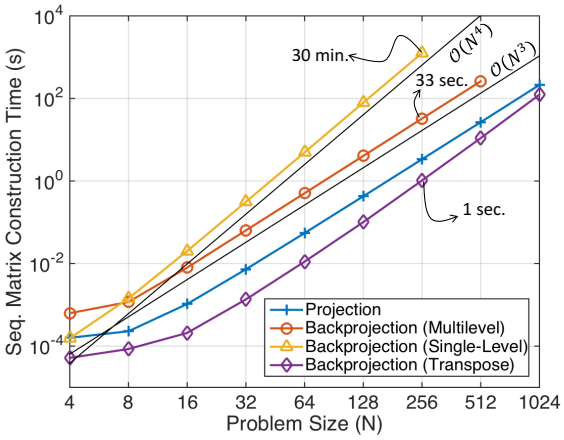

Fig. 2. Sequential construction times of projection and backprojection matrices. The multilevel construction scales with $\mathcal{O}(N^3)$. Constuction with matrix transposition also scales with $\mathcal{O}(N^3)$, however, it is not parallelizable.

Fig. 2 shows sequential reconstruction times of projection and bakcprojection matrices. In the experiments, the rays perform a raster scan with N projections and each projection involves N rays. Consequently, the number of rays is equal to the number of pixels, i.e., $M = N \times N$, so that the imaging problem is not under- or over-determined. The projection matrix is constructed by a ray-tracing algorithm and the backprojection matrix is constructed by three algorithms: matrix transpose, single-level (naive), and multilevel (proposed). The figure shows the scalings of single-level and multilevel algorithms are in a good agreement with $\mathcal{O}(N^4)$ and $\mathcal{O}(N^3)$ curves. For the case of $N = 256$, the naive and proposed algorithms take 30 minutes and 33 seconds, respectively, whereas the matrix transpose takes just a second. However, the traspose algorithm is not parallelizable and therefore is not suitable for a parallel implementation.

Fig. 3 shows the parallel matrix construction times for $N = 512$ on various number of cores. From sequential to 64 cores, OpenMP threads are employed on a single node. From 128 to 4,096 cores, MPIxOpenMP programming is used to distribute the threads among distributed-memory nodes, where each node is employed with 64 threads. The figure shows that the matrix construction speeds up 48.2x on 64 cores (with 75% efficiency) and 1247x on 4096 cores (with 30% efficiency).

The inefficiency at high number of cores is mainly due to the low granularity of the top levels. For example, level 3 has only 64 subdomains and therefore it can be parallelized over 64 cores at most. In this problem, the partitioning yields 10 levels, however, only the bottom three levels has enough subdomains to be parallelized among 4,096 threads.

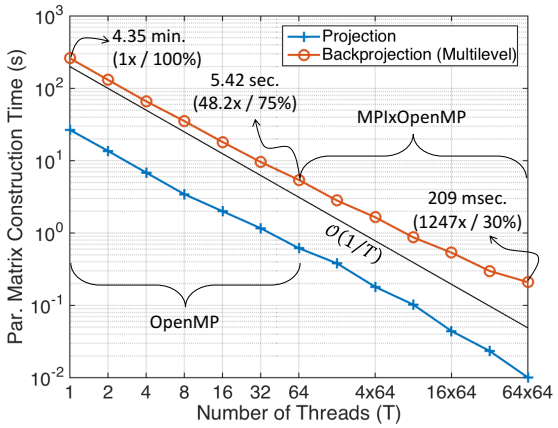

Fig. 3. Massively-parallel construction times of projection and backprojection matrices. Both constructions scales well up to 4,096 cores on 64 KNL nodes.

IV. Conclusions

For memory-bound iterative x-ray image reconstruction, the explicit sparse backprojection matrix is required. Naive methods for constructing the sparse matrix has either a high computational complexity or race conditions. For overcoming the bottleneck, a multilevel algorithm is proposed for low-complexity and massively parallel construction of the backprojection matrix. By a set of experiments, we demonstrate the $\mathcal{O}(NM)$ computational complexity and 1247x parallel speedup on 64 KNL nodes for the proposed algorithm.

Acknowledgment

This material is based upon work supported by the U.S. De-partment of Energy under Contract No. DE-AC02-06CH11357. M. Hidayetoğlu was a Givens associate at Argonne in Summer 2018.

References

[1] A. Buluç, *et al.*, "Parallel sparse matrix-vector and matrix-transpose-vector multiplication using compressed sparse blocks," *ACM Symposium on Parallelism in Algorithms and Architectures*, Aug. 2009, Calgary, Canada.

[2] Y.-D. Liang and B. A. Barsky, "A new concept and method for line clipping," *ACM Trans. Graph.*, vol. 3, no. 1, Jan. 1984, pp. 1–22.