Parallel Prefix Algorithms for the Registration of Arbitrarily Long Electron Micrograph Series

Extended Abstract

Marcin Copik
RWTH Aachen
mcopik@gmail.com

Paolo Bientinesi (advisor)
RWTH Aachen
pauldj@aices.rwth-aachen.de

Benjamin Berkels (advisor)
RWTH Aachen
berkels@aices.rwth-aachen.de

ACM Reference Format:
Marcin Copik, Paolo Bientinesi (advisor), and Benjamin Berkels (advisor). 2021. Parallel Prefix Algorithms for the Registration of Arbitrarily Long Electron Micrograph Series. In *Proceedings of Supercomputing 2017, Denver, Colorado, USA, 2017 (SC17),* 2 pages.
https://doi.org/10.1145/nnnnnnn.nnnnnnn

Modern electron microscopy allows observation of specimens at a nanometer resolution. However, the quality of acquired images might be limited by an interaction of the electron beam with the specimen, movement of the sample during the acquisition or other environmental factors. Berkels et.al.[1] proposed a new approach to increase the amount of information extracted from microscopy images. Instead of using a single high dose frame, a series of low dose noisy frames $f_0, f_1, \ldots, f_n$ is acquired. Afterwards, frames are aligned to the first image f_0 to represent only the physical change of the object and not the sample drift. The alignment is achieved by a two–step *registration* of images which estimates a rigid deformation $\phi_{0,i}$ such that $f_0 \approx f_i \circ \phi_{0,i}$.
The initial step applies a registration function to each pair of images (f_i, f_{i+1}), producing a sequence of deformations $\phi_{i,i+1}$. Then, a prefix sum is computed over deformations with a second registration function used as the sum operator. In our experiments, a serial registration of frames from ten seconds of microscopy acquisition requires almost thirteen hours of computation. Since the registration becomes impractical for longer series of frames, we intend to speedup this process by parallelizing it. Although the first step of registration is trivially parallelizable due to lack of dependencies, the prefix sum phase requires a more sophisticated approach.

Prefix sum has been researched for decades and it is considered to be a basic primitive for building parallel algorithms[2]. Multiple variants of a parallel prefix sum have been designed to decrease span, the length of a critical path in the algorithm, by performing more work, but in parallel. These algorithms differ in the efficiency of parallel execution, the amount of work performed and propagation of dependencies between iterations. We intend to use the parallel prefix sum as a basis for parallelization strategy of the image registration problem. However, properties of the image registration process are entirely different from prefix sum problems discussed in the literature. So far, the work has been focused on memory–bound operators where the cost of accessing and moving data is significantly higher than an application of the operator. In our problem, approximating a deformation takes several seconds. Furthermore, the iterative nature of registration does not allow to predict a total cost of computation, and we have observed huge variances in execution time between different pairs of frames.

For an arbitrarily long series of data acquisition, we need a distributed implementation of a parallel prefix sum to scale the computation across a cluster. The cost of each operator application is very high, thus we intend to minimize the span and maximize the parallelism within the algorithm. Since the execution time of an image registration is not known a priori, the safest option is to distribute images equally between MPI ranks. A dynamic schedulingb is not applicable because of the distributed environment and the algorithmic structure of prefix sum.
The algorithm[6] is split into three phases and it starts with a local sequential prefix sum, performed independently by each worker. Partial results from this step are used in a global parallel prefix sum and each rank obtains a sum of deformations on all preceding ranks. The second local phase updates partial prefix sums with this result. The span model of the algorithm proves that a linear scaling is not achievable, because of a redundant work performed to parallelize the execution. It provides us with an upper boundary on attainable speedup, which depends on the span of the global prefix sum.

The algorithm has been evaluated on the case study data representing aluminum oxidization process. Variants of global prefix sum include span–optimal Sklansky[5] and Kogge–Stone[4] prefix sums, a work–efficient Blelloch[3] algorithm, and an inclusive and exclusive scan implementations in OpenMPI and IntelMPI. Strong scaling results indicate that the maximum performance can not be achieved because of an ill-balanced workload. We have measured weak scaling as well, to verify if the algorithm is capable of solving larger problems with more hardware. The span model suggests that we can expect a constant increase in execution time, which should not depend on the initial data size or the number of processor cores. Similarly to the previous case, results do not meet theoretical estimations.
The efficiency of the algorithm is not high, mostly because of the non–ideal parallelization of prefix sum and load balancing issues. In many experiments, span–optimal prefix sums tend to provide the shortest execution time, but the difference between them and Blelloch or IntelMPI scan is not large. We conclude that the most promising way of increasing the efficiency and scaling is to focus on optimizing a multithreaded implementation of the image registration operator.

SC17, 2017, Denver, Colorado, USA
2021. ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

REFERENCES

[1] Benjamin Berkels, Peter Binev, Douglas A. Blom, Wolfgang Dahmen, Robert C. Sharpley, and Thomas Vogt. 2014. Optimized imaging using non-rigid registration. *Ultramicroscopy* 138 (2014), 46 – 56. https://doi.org/10.1016/j.ultramic.2013.11.007
[2] G. E. Blelloch. 1989. Scans As Primitive Parallel Operations. *IEEE Trans. Comput.* 38, 11 (Nov. 1989), 1526–1538. https://doi.org/10.1109/12.42122

[3] Guy E. Blelloch. 1990. *Prefix Sums and Their Applications*. Technical Report CMU-CS-90-190. School of Computer Science, Carnegie Mellon University.
[4] Peter M. Kogge and Harold S. Stone. 1973. A Parallel Algorithm for the Efficient Solution of a General Class of Recurrence Equations. *IEEE Trans. Comput.* 22, 8 (Aug. 1973), 786–793. https://doi.org/10.1109/TC.1973.5009159
[5] J. Sklansky. 1960. Conditional-Sum Addition Logic. *IRE Transactions on Electronic Computers* EC-9, 2 (June 1960), 226–231. https://doi.org/10.1109/TEC.1960.5219822
[6] Ömer Eğecioğlu, Cetin K. Koc, and Alan J. Laub. 1989. A recursive doubling algorithm for solution of tridiagonal systems on hypercube multiprocessors. *J. Comput. Appl. Math.* 27, 1 (1989), 95 – 108. https://doi.org/10.1016/0377-0427(89)90362-2 Special Issue on Parallel Algorithms for Numerical Linear Algebra.