Poster: GPU-accelerated rigid body fitting of atomic structures into electron density maps

Edward W. Lowe Jr., Nils Woetzel, Jens Meiler
Department of Chemistry,
Center for Structural Biology
Vanderbilt University
Nashville, TN, USA
jens@jens-meiler.de

Keywords--GPU, OpenCL, cryo-EM, fitting, density map

I. INTRODUCTION

Many protein structures are elucidated to atomic detail using x-ray crystallography or nuclear magnetic resonance (NMR) experiments and are deposited in the PROTEIN DATA BANK (PDB). However, macromolecular assemblies are most relevant in understanding biological processes. Cryo-electron microscopy (cryo-EM) is an experimental technique that generates 3D electron density maps of medium resolution (7-10Å) for large macromolecular assemblies [1]. A fitting or docking procedure for known atomic models within the assembly is needed to place these models into the experimental density map. Initial likely placements are refined to find the local optimum and discard incorrect arrangements [2].

In the last three years, graphical processing units (GPUs) have grown in popularity since they can now be programmed with higher-level languages like the industry standard OpenCL [3]. GPUs are ideal in processing parallel computations and show significant speed ups over CPU implementations [4]. The presented work applies GPUs to rapidly refine initial placements of protein models into low resolution density maps thereby accelerating construction of atomic-detail models for large macromolecular assemblies.

II. RESULTS

Three initial fits of 1ubi in a 6.6Å resolution synthesized density map had backbone RMSDs to the correct placement of 2.7, 2.9 and 6.6 Å. They have been refined with a Powell optimizer [5] in 10 iterations using 6 directions, 3 rotations α, β with 0.15 radians and γ with 0.075 radians starting direction to cover the entire euler sphere and translation with 1.1 Å search directions in x, y and z. Each line search was terminated after 10 iterations or if *d*CCC < 0.001. TABLE I. summarizes timings on CPU and 4 different GPU architectures in double floating point precision.

TABLE I. TIMINGS FOR POWELL OPTIMIZATIONS IN SECONDS

	CCC		Xeon	NVIDIA			ATI
	Start	Final	W3570 3.2GHz	Tesla C1060	Tesla C2050	GTX 470	Radeon HD 5970
1	0.892	0.989	38.486	7.260	1.423	2.398	2.486
2	0.862	0.989	35.579	6.944	1.468	2.138	2.443
3	0.698	0.842	23.888	7.151	1.476	2.184	2.518

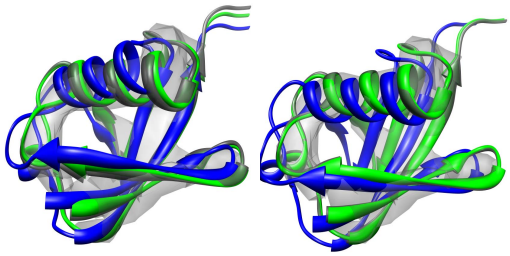

Figure 1. Initial (blue) and refined (green) placements of 1ubi for initial fit 1 (left) and initial fit 2 (right). The optimal placement is shown in grey. A 6.9 Å resolution density map simulated using Colores is shown as transparent envelope.

III. DISCUSSION

Density map and coordinate handling is highly parallelizable by nature due to the grid layout of the density map and the basic vector matrix algebra involved in coordinate transformation. The minimization using a Powell optimizer with Golden section line search a GPU speed up compared to a CPU implementation by a factor of 27 was achieved, yielding identical results.

The program is available free of charge for non-commercial use under: http://www.meilerlab.org and http://bclcommons.vueinnovations.com/bclcommons. The OpenCL kernel sources are bundeled and compiled just-in-time with the program start.

REFERENCES

[1] J. Frank, "Single-particle reconstruction of biological macromolecules in electron microscopy--30 years.," *Quarterly reviews of biophysics*, vol. 42, Aug. 2009, pp. 139-58.

[2] W. Wriggers, R.A. Milligan, and J.A. McCammon, "Situs: a package for docking crystal structures into low-resolution maps from electron microscopy," *Journal of Structural Biology*, vol. 125, 1999, p. 185–195.

[3] J.E. Stone, D. Gohara, and G. Shi, "OpenCL: A Parallel Programming Standard for Heterogeneous Computing Systems.," *Computing in science & engineering*, vol. 12, May. 2010, pp. 66-72.

[4] J. Cohen and M. Garland, "Novel Architectures: Solving Computational Problems with GPU Computing," *Computing in Science & Engineering*, vol. 11, Sep. 2009, pp. 58-63.

[5] M.J.D. Powell, "An efficient method for finding the minimum of a function of several variables without calculating derivatives," *The Computer Journal*, vol. 7, Feb. 1964, pp. 155-162.