nuFFTW: A Parallel Auto-Tuning Library for Performance Optimization of the nuFFT

Mark Murphy[1,3], Michal Zarrouk[1,2], Kurt Keutzer[1], and Michael Lustig[1]

[1]EECS, UC Berkeley, Berkeley, CA, United States,

[2]Advanced Light Source, Lawrence Berkeley National Laboratory, Berkeley, CA, United States, [3]Google, Mountain View, CA, United States

Purpose

Develop an auto-tuned fast parallel library to compute the non-uniform Fast Fourier Transform (nuFFT)[1,2]. Leverage auto-tuned FFT and sparse matrix multiplication libraries in addition to auto-tuning over algorithm parameters for the fastest implementation tailored to the architecture.

Theory

•Gridding-based nuFFT implementations span a spectrum of precomputation levels (fig. 1), with a corresponding tradeoff between arithmetic throughput and memory usage\transfer rates.

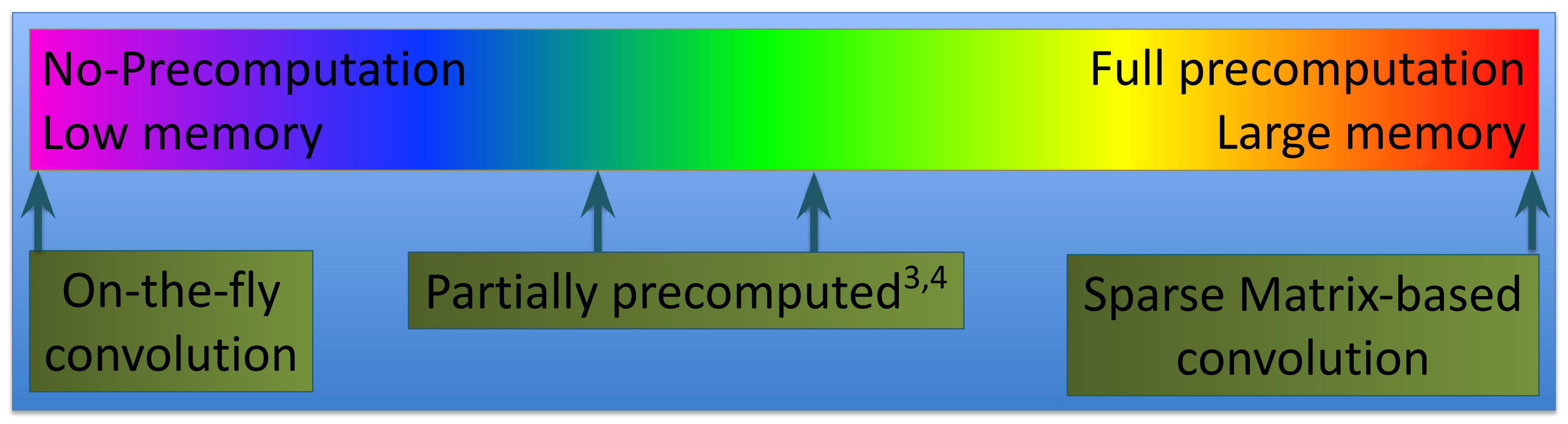

***Figure 1:** nuFFT resampling Implementation tradeoffs*

•Influence of nuFFT implementation and parameter selection for a given hardware on resulting runtime is non-trivial.

•The maximum aliasing amplitude (MAA)[5] defines a space of error-equivalent pairs of kernel widths and grid oversampling ratios (fig. 2).

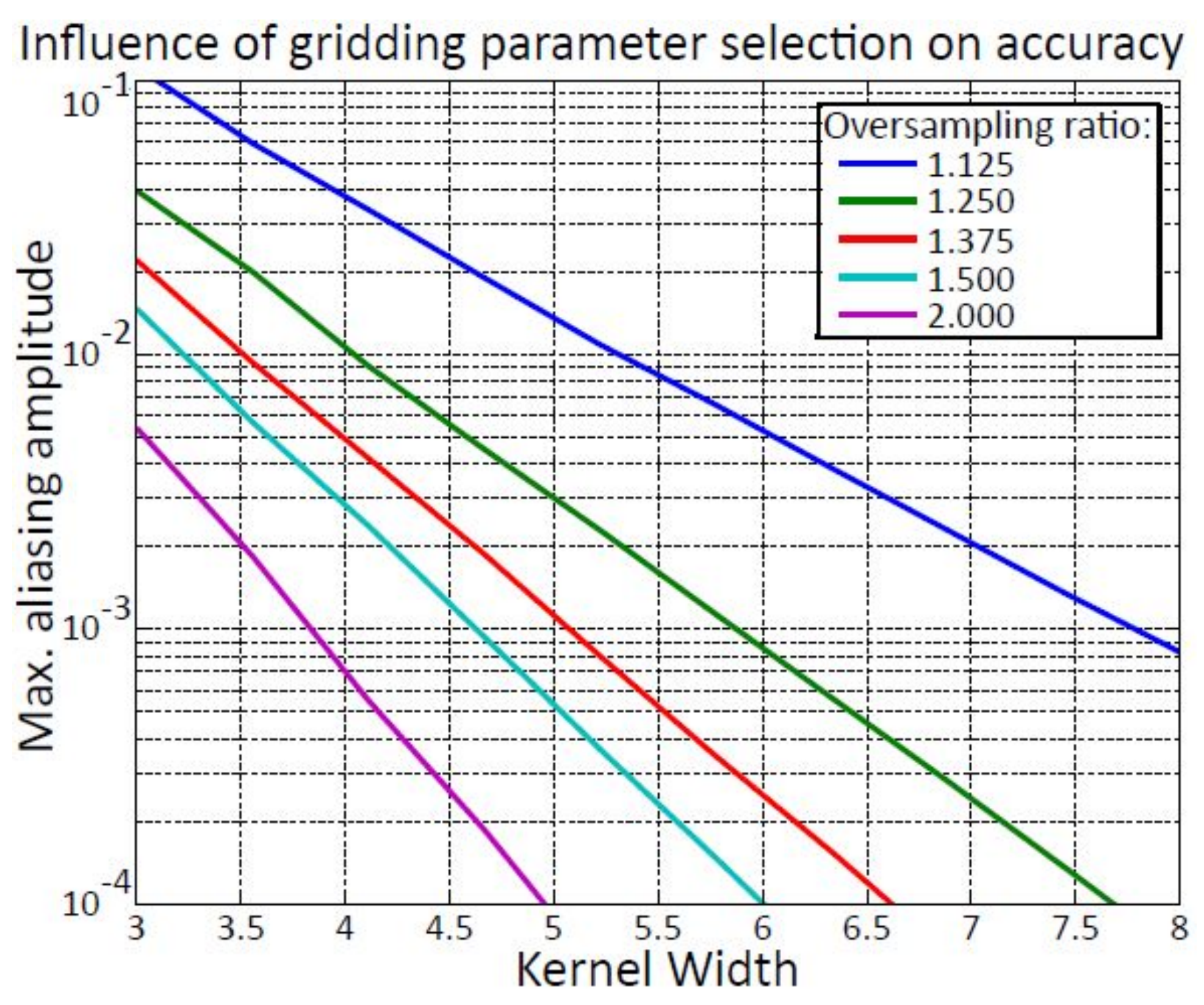

***Figure 2:** Maximum aliasing amplitude[5] of various gridding parameter selections: grid oversampling ratio and interpolation kernel width.*

Auto-tuning the nuFFT

•We present a fast, auto-tuned, Gridding-based nuFFT library with parallel implementations on CPUs and GPUs for reconstructing from non-equispaced k-space data.

•Our auto-tuning approach empirically selects an optimal implementation per trajectory by searching over algorithms and parameters, and saves it for future reconstructions.

•Currently our library implements both ends of the precomputation spectrum, however it could be expanded to include any method of nuFFT computation.[7,8,9,10]

Simplified tuning heuristic

•As exhaustive search is sometimes expensive, we propose a simplified heuristic where only runtime of the FFT phase is minimized. Since FFT runtimes don't depend on the gridded trajectory, this heuristic requires only a one-time FFT benchmark during system installation.

Methods and results

Performance results were measured on a 12-core 2.67 GHz Intel Westmere CPU and Nvidia GTX580 GPU. We use the Kaiser-Bessel kernel[1] with parameters chosen to satisfy a MAA=10^{-2}. We rely on external optimized and auto-tuned libraries for sparse matrix operations (OSKI[6] and CUSparse) and FFT (FFTW and CUFFT). Gridding was performed on Cones trajectories with isotropic 1mm spatial resolution.

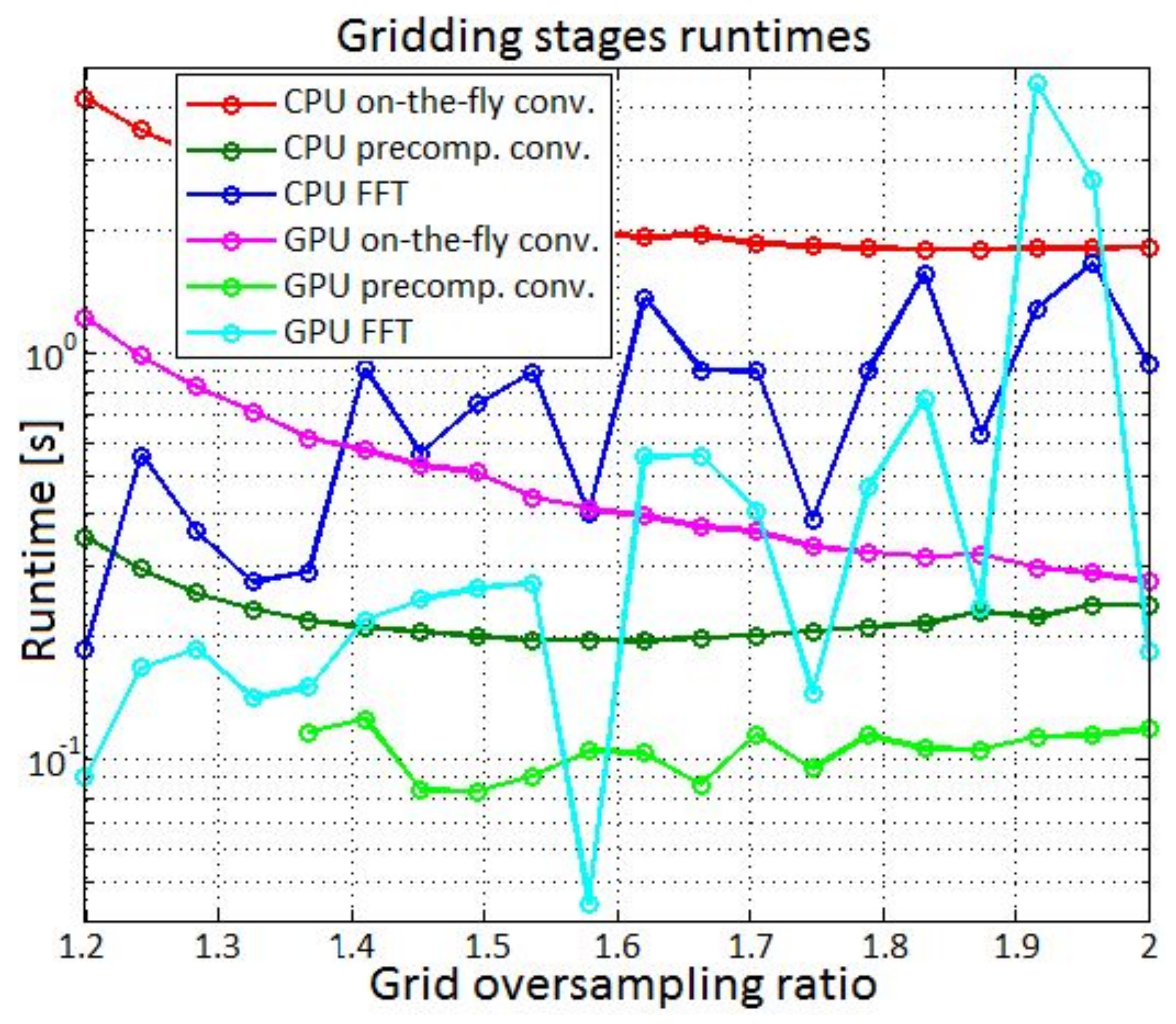

***Figure 3:** Runtimes of various gridding phases, for a 14M-sample Cones trajectory, on CPU and GPU, for different selections of grid oversampling ratio.*

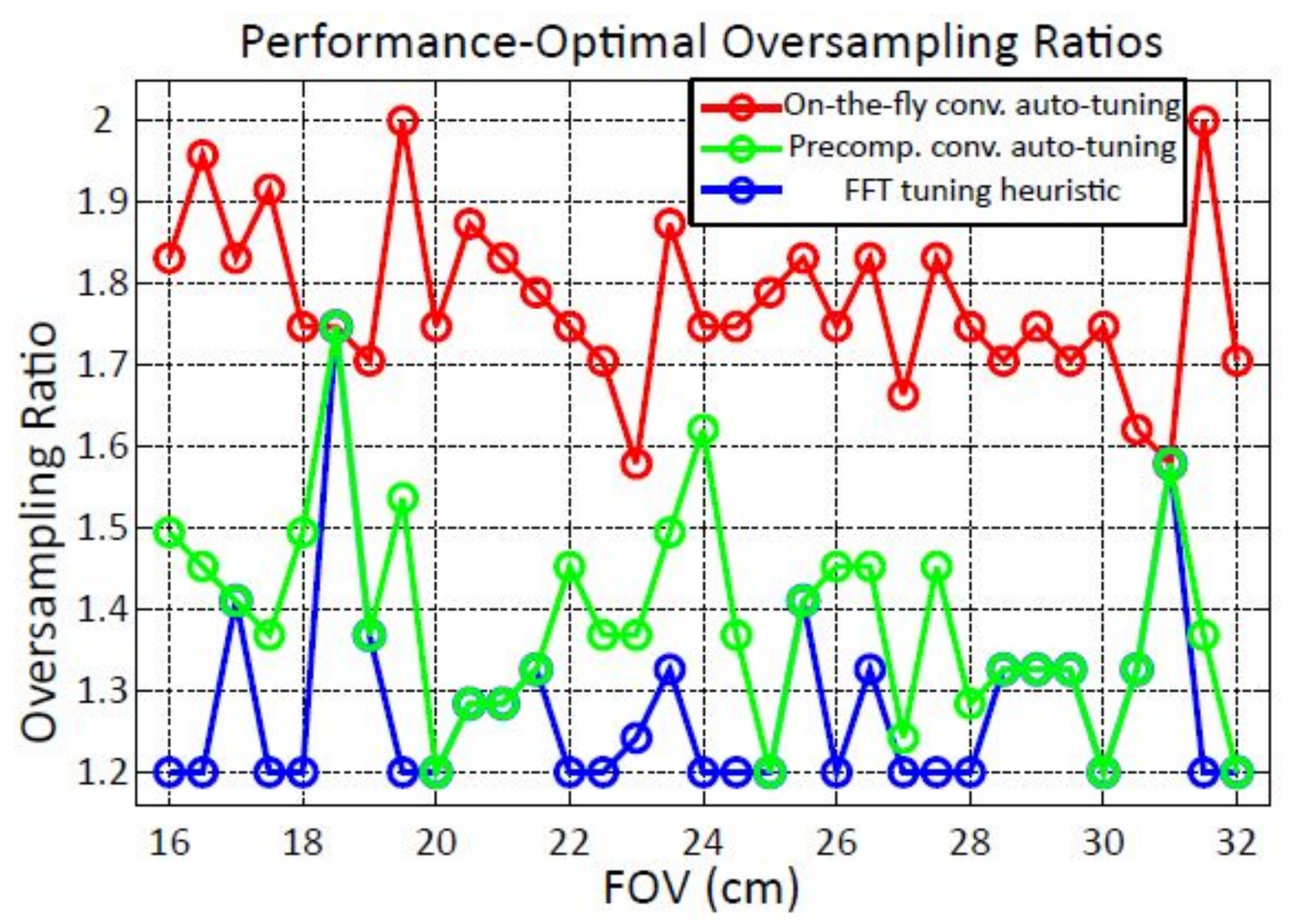

***Figure 4:** Grid oversampling ratios selected by various auto-tuning heuristics, for a range of FOVs.*

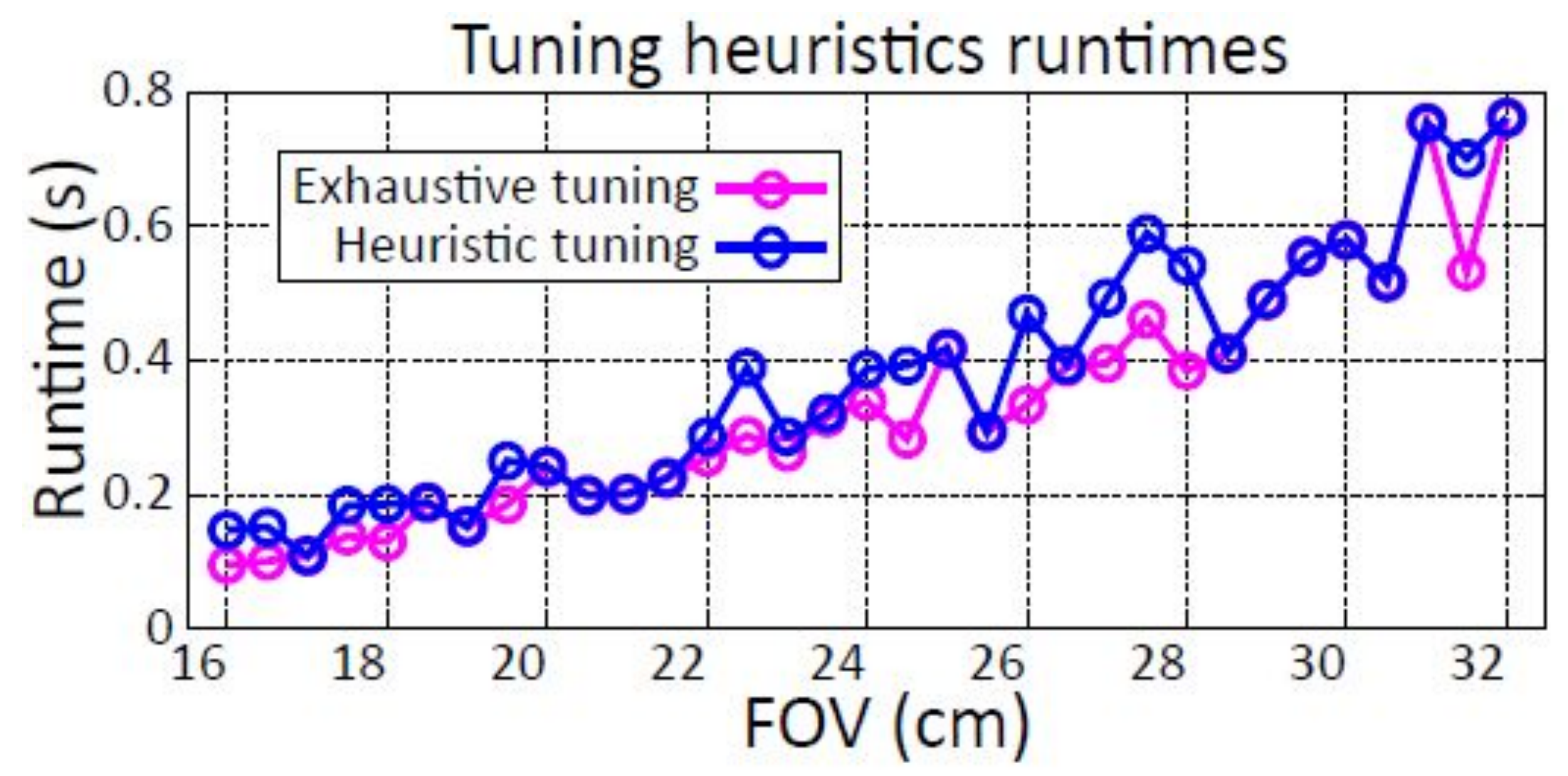

***Figure 5:** Optimized runtimes of various tuning heuristics, for a range of FOVs.*

•Precomputation-based convolution is faster than on-the-fly convolution. However, due to higher throughput of GPUs over CPUs (relative to memory bandwidth), the performance acceleration on GPUs is lower than CPUs.

•Precomputation-based convolution was infeasible on GPU for small grid oversampling ratios which resulted sparse matrices larger than available memory on the GPU. Best FFT performance is achieved when grid size factors into small prime numbers.

Conclusions

•Gridding runtimes depend greatly on the implementation parameters, but also on the underlying architecture, available memory and the sampled trajectory.

•Auto-tuning the nuFFT will speed up MR image reconstruction times. Empirical optimization enables accounting for micro-architectural changes in future processor generations.

•Complete gridding parameter tuning is particularly beneficial when a trajectory is to be used many times for image reconstruction (multiple receivers, dynamic imaging).

•When offline tuning is impractical, our suggested heuristic of tuning the FFT phase achieves near-optimal performance.

1. Jackson et al. *IEEE Trans. Med. Imag.* 1991 Sep;10(3):473–478.
2. M. Murphy. PhD Thesis, EECS, UC Berkeley. 2011.
3. Sorensen et al. *IEEE Trans. Med. Imag*. 2008 Apr;27(4):538-547.
4. Obeid et al. *Proc. ISMRM*. 2011;2547.
5. Beatty et al. *IEEE Trans. Med. Imag*. 2005 Jun;24(6):799-808.
6. Vuduc et al. *Proc. SciDAC, J. Physics: Conf. Ser.* 2005 Jun;16:521-530.
7. Nam et al. *Proc. ISMRM*. 2011;2548.
8. Wu et al. *Proc. ISMRM*. 2011;4396.
9. Keiner et al. *ACM Trans. Math. Software*. 2009;36(19):1-30.
10. Fessler et al. *IEEE Trans. Sig. Proc.* 2003 Feb; 51(2):560-574.

Acknowledgment: The work of MZ is supported by Lawrence Berkeley National Laboratory.