Cache Oblivious Fast 2D Fourier Transform*

G.Y. Zavyalov, A.S. Levin, M.E. Beketov

HSE University

Fourier transform (FT) is a fundamental operation, providing basis for numerous methods in pure and applied mathematics. The celebrated Fast Fourier Transform (FFT) algorithm of Cooley and Tukey [1] enables efficient computation of discrete FT (DFT) of 1-dimensional signals. When the dimension is higher, in practice DFT is typically done by running FFT along each dimension (axis) of the signal array. While in fact one can explicitly generalize the idea behind FFT to 2+ dimensions: this is known as vector-radix FFT algorithm [2, 3], recently rediscovered [4]. Despite the 25% reduction of required arithmetic operations (complex multiplications), practical implementations of this algorithm are lacking in popular software libraries specializing in FTs.

In present work we study space complexity and cache behavior of said algorithm (which we refer to as F2FT for Fast 2D Fourier Transform), and show that it is in fact cache-oblivious [5], a remarkable property meaning that the number of cache misses is independent of the cache memory architecture. We provide our own C++ implementation of F2FT, which utilizes a custom layout of the processed arrays in memory (based on bit-reversal techniques). We benchmark our implementation versus that found in PocketFFT C++ library (a development over the FFTPack library [7]), as used in SciPy [6] of version 1.13.1. In PocketFFT, 2D DFT is achieved by separately doing FFT along two axes. We find a noticeable advantage of our implementation in terms of runtime and (write) cache misses – run on CPUs of both ARM64 and x86_64 architectures (in case when all arrays fit into RAM).

References

1. Cooley J., Tukey J. An algorithm for the machine calculation of complex Fourier series // Mathematics of computation. 1965. Vol. 19, no. 90. P. 297–301. DOI: 10.2307/2003354.

2. Rivard G. Direct fast Fourier transform of bivariate functions // IEEE Transactions on Acoustics, Speech, and Signal Processing. 1977. Vol. 25, no. 3. P. 250–252. DOI: 10.1109/TASSP.1977.1162951.

3. Harris D., McClellan J., Chan D. et al. Vector radix fast Fourier transform // ICASSP'77. IEEE International Conference on Acoustics, Speech, and Signal Processing. 1977. Vol. 2. P. 548–551. DOI: 10.1109/ICASSP.1977.1170349.

4. Tutatchikov V. Two-dimensional fast Fourier transform: Batterfly in analog of Cooley-Tukey algorithm // 11th International Forum on Strategic Technology (IFOST). 2016. P. 495–498. DOI: 10.1109/IFOST.2016.7884163.

5. Frigo M., Leiserson C., Prokop H. Cache-oblivious algorithms // 40th Annual Symposium on Foundations of Computer Science. 1999. P. 285–297. DOI: 10.1109/SFFCS.1999.814600.

6. Virtanen P., Gommers R., Oliphant T. et al. SciPy 1.0: fundamental algorithms for scientific computing in Python // Nature methods. 2020. Vol. 17., no. 3. P. 261–272. DOI: 10.1038/s41592-019-0686-2.

7. Swarztrauber P. Vectorizing the FFTs // Parallel computations. 1982. P. 51–83. DOI: 10.1016/B978-0-12-592101-5.50007-5.

*The work of M. Beketov was supported by HSE University Basic Research Program.