Hybrid CPU-GPU computing for sparse signal processing

Context

Sparse signal processing is about exploiting the sparse nature of signals to process them more efficiently or to measure them to some prescribed level of accuracy with fewer measurements than those conventional methods require [1]. The idea is that we would like to retrieve some signal belonging to a high dimensional space while leveraging its sparse nature so as to lower the number of measurements needed to reconstruct it in comparison to traditional approaches (which require at least as many measurements as space dimensions). Among the classes of sparse recovery algorithms, greedy ones are particularly important for embedded applications because of their low space and time complexity (in comparison to other alternatives based on convex optimization problems) [2]. A particular class of greedy algorithms is that tailored to multiple measurement vector (MMV) problems: several signals are acquired by different acquisition channels, which may be different but, by their nature, share the same support (i.e., the position of their non-zero entries are identical or, at least, similar). These problems appear in many practical applications, including some where algorithms are implemented on embedded platforms with limited computational capabilities [3-6].

There has been some research on the computationally optimal implementation of greedy algorithms such as orthogonal matching pursuit (OMP). Some of this research is mathematical (see, e.g., [2]) and aims to find algebraic ways of recasting the original OMP algorithm into one with lower complexity in time and space. Other avenues of optimization focus on how to optimally implement such algorithms on hardware platforms: regular processors, processors augmented with graphical processing units (GPUs), field programmable gate arrays (FPGAs) (e.g., [7]) and application-specific integrated circuits (ASICs) [8]. Surprisingly, despite their high costs, most research on the subject has focused on FPGA and ASIC-based architectures (see, e.g., [9, Sec. 4.2, Table 3] for a recent review).

Objectives and steps

Loosely speaking, this master's thesis studies how to best implement OMP, particularly in multiple measurement vector (MMV) scenarios. In particular, the master's thesis student is expected to evaluate to what extent GPU-based computations are faster or slower than those on ARM or x86_64 processors, depending on the number of measurement vectors in the MMV problem. Steps are: *i)* to review mathematically optimized implementations of OMP, *ii)* to program a naïve OMP in Python or Matlab to get an idea of how it works, *iii)* to implement OMP for MMV problems using BLAS/LAPACK (with an MKL or OpenBLAS implementation) and/or Eigen, *iv)* to implement OMP for MMV problems using CUDA (and libraries such as cuBLAS and cuSOLVER when appropriate) and *v)* to compare results against the naïve Python/Matlab implementation and other existing implementations if they are available as well as to determine the extent to which GPU-based algorithms run faster or slower than those limited to running on CPUs only. The BEAMS-EE department possesses various CPU-GPU platforms that the student can use (ranging from embedded devices to a server rack with dual GPUs).

Student profile

This subject is appropriate for students willing to translate mathematical algorithms into efficient implementations running on modern computing architectures. It requires some skills in linear algebra, computer architecture and C/C++/CUDA programming. Having followed the course "Microprocessor architecture" and/or a course on GPU programming is a plus but is not mandatory. Having experience with BLAS/LAPACK is a plus as well.

References

[1] Donoho, D. L. (2006). Compressed sensing. IEEE Transactions on information theory, 52(4), 1289-1306.
[2] Sturm, B. L., & Christensen, M. G. (2012, August). Comparison of orthogonal matching pursuit implementations. In 2012 Proceedings of the 20th European Signal Processing Conference (EUSIPCO) (pp. 220-224). IEEE.
[3] Mishali, M., Eldar, Y. C., & Elron, A. J. (2011). Xampling: Signal acquisition and processing in union of subspaces. IEEE Transactions on Signal Processing, 59(10), 4719-4734.
[4] Mishali, M., & Eldar, Y. C. (2009). Blind multiband signal reconstruction: Compressed sensing for analog signals. IEEE Transactions on signal processing, 57(3), 993-1009.
[5] Shahmansoori, A., Garcia, G. E., Destino, G., Seco-Granados, G., & Wymeersch, H. (2017). Position and orientation estimation through millimeter-wave MIMO in 5G systems. IEEE Transactions on Wireless Communications, 17(3), 1822-1835.
[6] Zhou, Y., Song, A., Tong, F., & Kastner, R. (2018). Distributed compressed sensing based channel estimation for underwater acoustic multiband transmissions. The Journal of the Acoustical Society of America, 143(6), 3985-3996.
[7] Huang, G., & Wang, L. (2017). An FPGA-based architecture for high-speed compressed signal reconstruction. ACM Transactions on Embedded Computing Systems (TECS), 16(3), 1-23.
[8] Chen, K. T., Ma, W. H., Hwang, Y. T., & Chang, K. Y. (2020). A low complexity, high throughput DoA estimation chip design for adaptive beamforming. Electronics, 9(4), 641.
[9] Djelouat, H., Amira, A., & Bensaali, F. (2018). Compressive sensing-based IoT applications: A review. Journal of Sensor and Actuator Networks, 7(4), 45.

Contact

Jean-François Determe, jean-francois.determe@ulb.be

Solbosch campus, building U, level. 5, BEAMS-EE department