Beamforming Adaptive Acoustic Arrays with Graphics Processing Units

Michael Romer
Department of Computers Sciences, Applied Research Labs
University of Texas at Austin

Mission

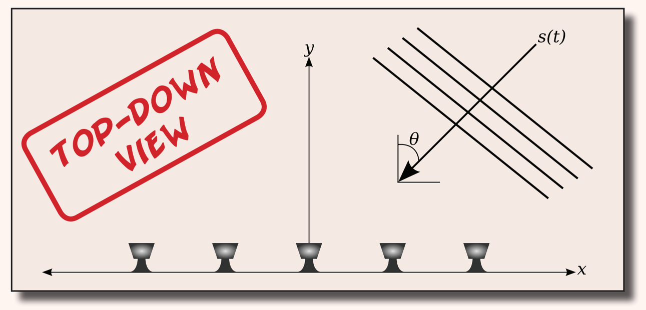
To determine the utility of graphics processing units as a means of hardware acceleration for adaptive beamforming in the context of US Naval applications.

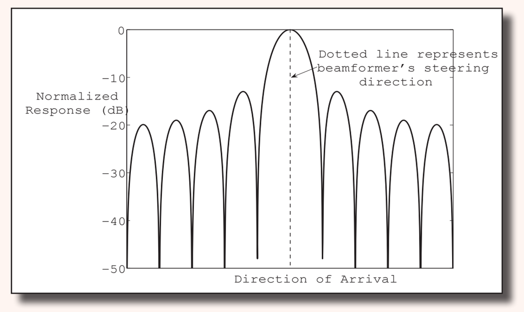
What is Beamforming?

Beamforming is a technique for determining the origin of a signal using an *array* of acoustic sensors. Each sensor is *omnidirectional*, i.e., records acoustic data (e.g., signal amplitude over time) equally from all directions.

...And How Does It Work?

A signal propagates towards the array of acoustic sensors from a *direction of arrival* (DOA):



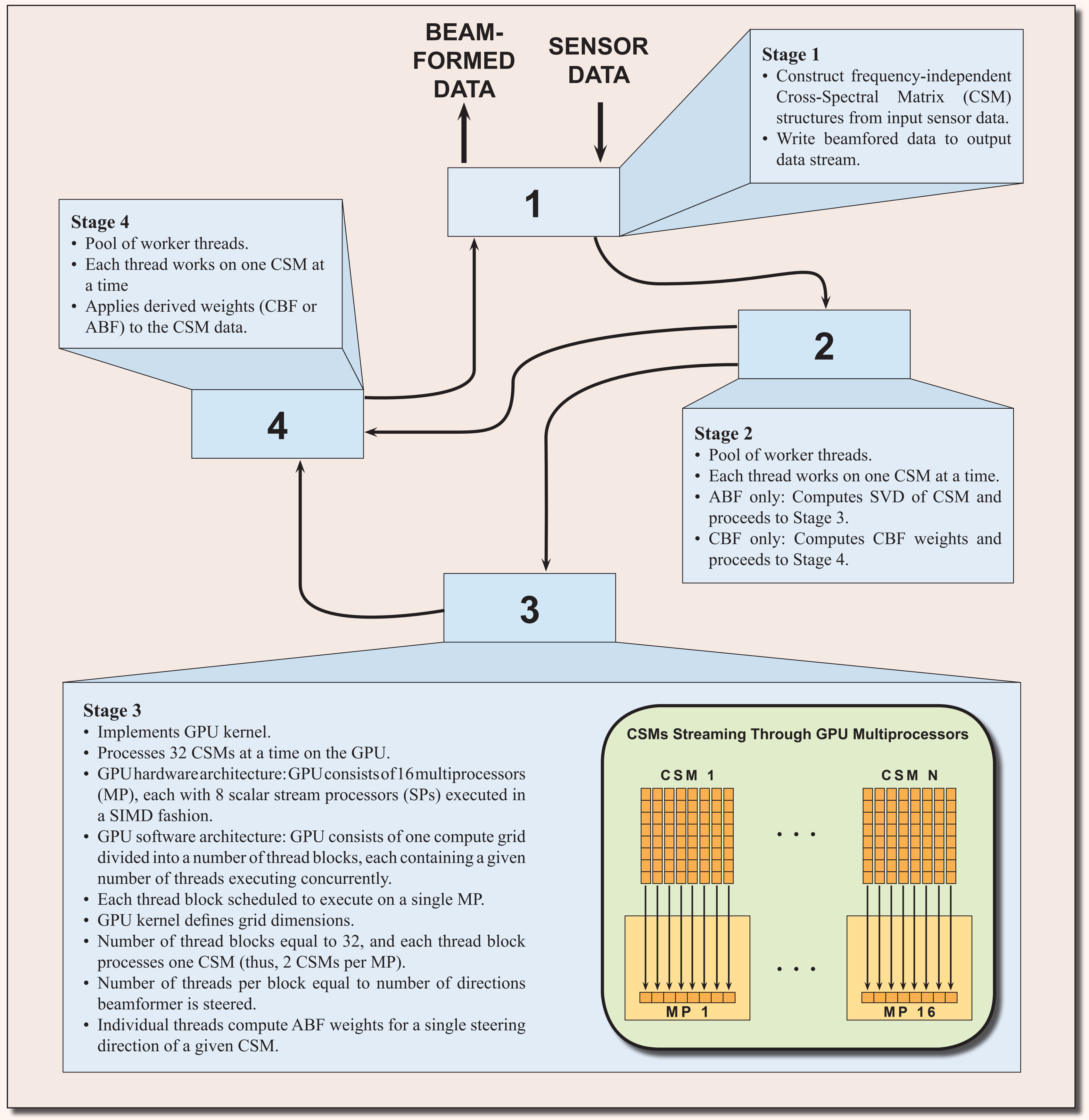


The beamformer weights the output of each sensor and sums all outputs together. The output of the beamformer is the *response*, and for non-adaptive methods, the response is maximal in the *steering direction* associated with the weights applied. Computing adaptive weights is computationally expensive and must be derived on a per frequency, per direction basis. The optimal weights will yield a steering direction that coincides with the observed signal's DOA.

Acknowledgements

I would like to thank Dr. Bill Mark and Dr. Don Fussell for being my computer science faculty advisors. Special thanks goes out to ARL for providing the IR&D funding for this project, and über mad props goes out to my supervisor Greg Thomsen for all the time he's dedicated in helping me make this project happen and all the crap he's had to tolerate because of it.

A GPU-Centric, Multithreaded Pipeline Architecture!



What's So Great About GPUs?

Technology	Performance (GFLOPS)	Reprogram-mability	Cost	Power Consumption	Form Factor
Application Specific Integrated Circuits (ASICs)	30 - 50 average	None	Very expensive	Very low	Very small
Field Programmable Gate Arrays (FPGAs)	20 average	Hardware Design Languages (HDL), Verilog	Moderate	Low	Small
General-purpose Processors (GPPs)/Clusters	50 peak	C, C++, Java, etc.	Relatively inexpensive	High	Large
Graphics Processing Units (GPUs)	330 peak	Shader languages, C with extensions (e.g., CUDA)	Cheap	Moderate	Moderate

GPUs provide a cheap parallel computing device that potentially provide higher performance per watt and performance per volume than any other means of hardware acceleration for adaptive beamforming.

Initial Results & Discussion

Compared to the reference Matlab implementation, the parallel GPU-based implementation yields a 2X speed-up; however, a serial implementation written in C resulted in a 4X speed-up. This factor of two slowdown can be attributed to the following:

- Lack of highly optimized linear algebra libraries for the GPU, particularly a singular value decomposition (SVD) solver.
- GPU kernel makes no use of on-chip shared memory. Minimizing accesses to global GPU memory and effective use of shared memory on each multiprocessor can lead to approximately an order of magnitude speed-up.
- Pipeline requires more load balancing in order to minimize stalls at each stage.

References

1. H. Krim and M. Viberg, "Two decades of array signal processing research," *IEEE Signal Processing Magazine*, vol. 13, no. 4, pp. 67-94, Jul. 1996.
2. H. Cox, R. Zeskind, M. Owen, "Robust adaptive beamforming," *IEEE Trans. Acoustics, Speech, and Signal Processing*, vol. 35, no. 10, pp. 1365-1376, Oct. 1987.
3. R.L. Walke, R.W.M. Smith, and G. Lightbody, "Architectures for adaptive weight calculation on ASIC and FPGA," in *Conf. Rec. Thirty-Third Asilomar Conf. Signals, Systems, and Computers*, 1999, pp. 1375-1380.
4. NVIDIA Corp. (2007, November). *NVIDIA CUDA Compute Unified Device Architecture: Programming guide* [Online]. Available: http://developer.download.nvidia.com/compute/cuda/1_1/NVIDIA_CUDA_Programming_Guide_1.1.pdf