5D interpolation on GPUs using the non-uniform Discrete Fourier Transform

Kai Zhuang* and Daniel Trad
kzhuang@ucalgary.ca

Abstract

We implement 5D interpolation with the Non-Uniform exact location DFT operator on GPUs to address binning-related errors in far offset approximations without compromising computational efficiency. This approach aims to accelerate the relatively slow Non-Uniform DFT operator using GPU computation. Our findings indicate that the GPU implementation effectively resolves both issues. Although the computation of the Non-Uniform DFT is more expensive than the FFT, transitioning to GPU computation significantly reduces runtime compared to the standard CPU implementation of DFT interpolation.

Introduction

Interpolating in 5D presents a formidable computational challenge due to the large data volumes in contemporary seismic acquisitions, often reaching multiple terabytes. This process involves reading and working with every point in the dataset multiple times, consuming a significant amount of computational time. It is not uncommon for days of server time to be dedicated to 5D interpolation to enhance sampling information and ensure data uniformity. By implementing a 5D interpolation workflow on GPUs, our goal is to bring the runtime of the Discrete Fourier Transform (DFT) on the GPU to a level that can match the speed of the Fast Fourier Transform (FFT) on a CPU. The inherent parallel nature of GPU processors, in contrast to the more serial processing nature of CPUs, provides an opportunity to make the DFT computation competitive with the Fast Fourier Transform (FFT) on a CPU.

Methods

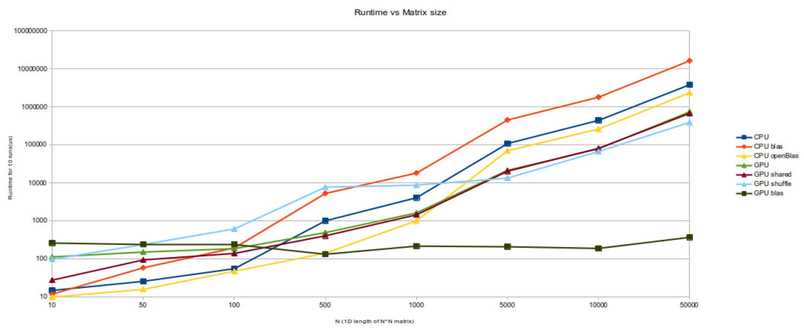

Figure 1: Log scale runtimes.

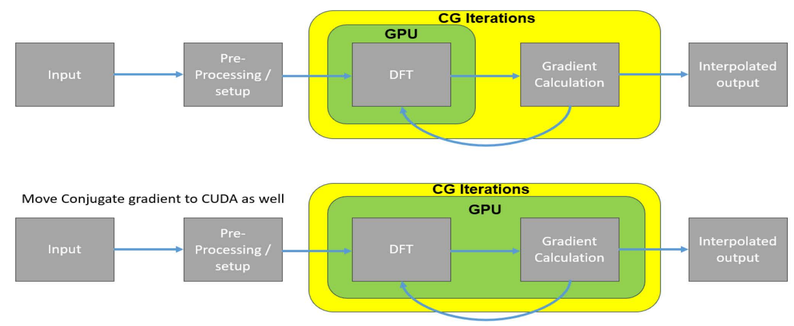

Figure 2: Ways to perform CUDA interpolation.

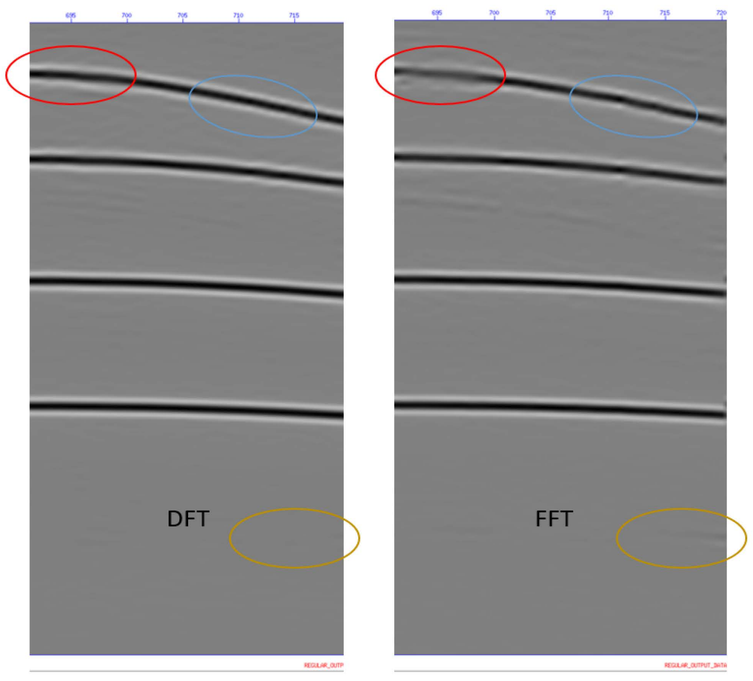

Figure 3: Zoomed difference between DFT and FFT.

Results

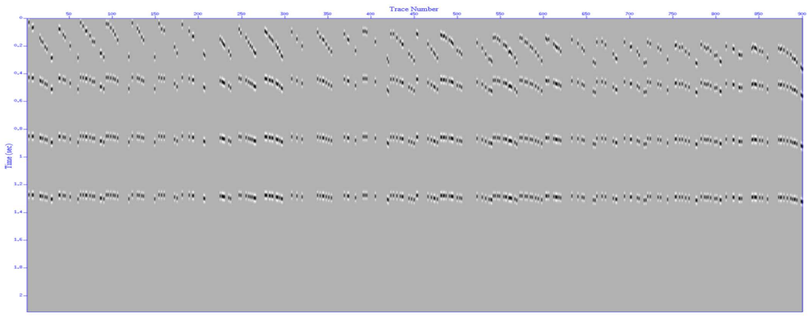

Figure 4: Decimated shot.

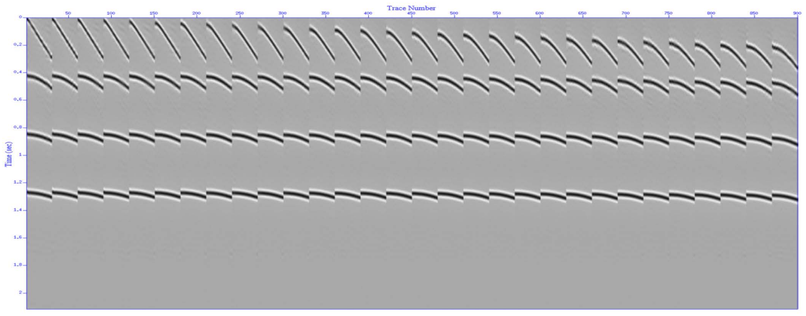

Figure 5: 5D interpolation using the DFT operator on CPU.

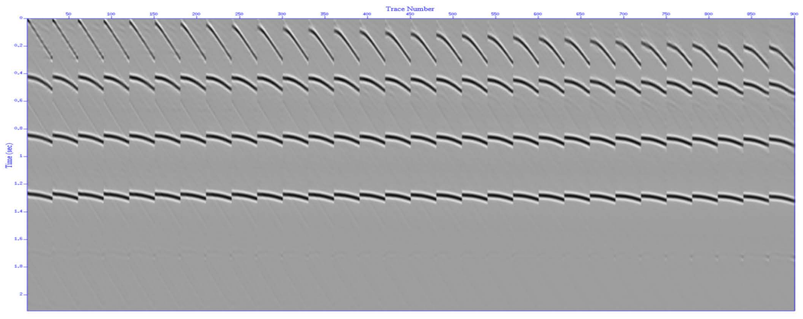

Figure 6: 5D interpolation using the DFT operator on GPU.

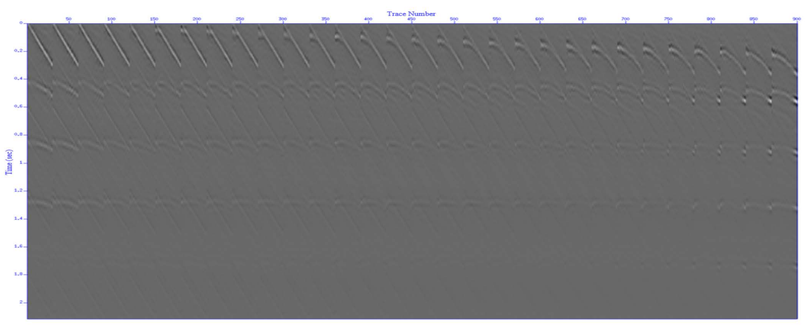

Figure 7: Difference between GPU and CPU operator results.

Acknowledgments

The sponsors of CREWES are gratefully thanked for continued support. This work was funded by CREWES industrial sponsors, NSERC (Natural Science and Engineering Research Council of Canada) through the grant CRDPJ 543578-19.

NSERC CRSNG
UNIVERSITY OF CALGARY
FACULTY OF SCIENCE
Department of Earth, Energy, and Environment