An OpenCL design for tomographic reconstruction on FPGA. Innovate Europe Design Contest Winners

Maxime Martelli, Mickael Seznec, Nicolas Heemeryck, Nicolas Gac

An OpenCL design for tomographic reconstruction on FPGA. Innovate Europe Design Contest Winners

Maxime MARTELLI, Mickael SEZNEC, Nicolas HEEMERYCK

TSA & L2S & SATIE, Telecom ParisTech, ECE

Industrial Context

1. With the **end of Moore's Law**, the semi-conductor industry seeks a reliable way to pursue the performance improvements of the last decades, and architecture-algorithm adequation is a solution for this new landscape.
2. Manufacturers like Intel and Xilinx are pushing for an FPGA resurgence, offering software suites and FPGAs card focused on a **software-like FPGA programming model**[1].

3D Computed Tomography Projection

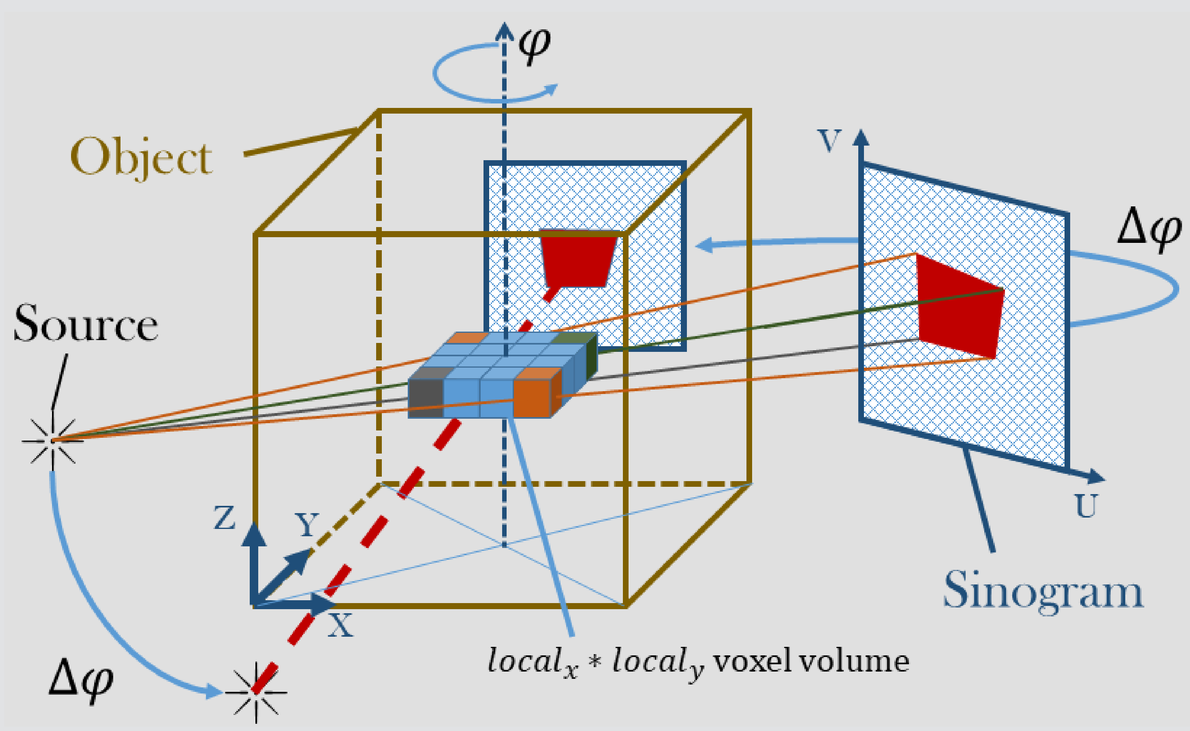


Figure 1: Figure caption

To compute the density of a given VOlume piXEL (voxel) $\vec{c} = (x, y, z)$, we sum up the contribution of every elementary detector (u, v) in line with the source and the considered voxel for every φ value as follows :

$$d(\vec{c}) = \int_0^{2\pi} s_{CT}(u(\varphi, \vec{c}), v(\varphi, \vec{c}), \varphi).w(\varphi, \vec{c})^2 \mathrm{d}\varphi \quad (1)$$

where $(u(\varphi_0, \vec{c}), v(\varphi_0, \vec{c}))$ are the values on the sinogram of the beam passing through $\vec{c}$ for $\varphi = \varphi_0$, and $w(\varphi_0, \vec{c})$ is the distance weight.[2]

OpenCL Work-group mechanism

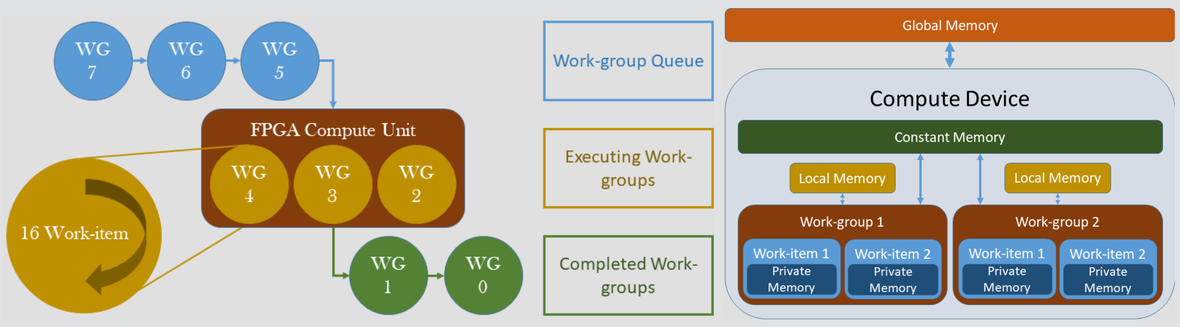


Figure 2: Work-group queueing mechanism and OpenCL Memory Architecture

There are two OpenCL kernel (device functions) programming models : **NDRange** (ND), similar to GPU programming and **single work-item** (SWI), similar to CPU programming. OpenCL allows work-group and work-item partitioning, as shown in Fig.2, and the challenge is to effectively use the inherent mechanism of each memory level to accelerate our design. This repartition allows effective internal communication and data synchronization between work-items of the same work-group.

Memory Benchmark

For benchmarking purposes, we implemented a custom routine program to measure the mean latency of each memory type on an Altera Cyclone V FPGA.

Memory structure	Mean latency (cycles)
Global	240
Constant	10
Local	15
Private	2

Table 1: Memory structure latency on an Altera Cyclone V.

Implemented OpenCL optimizations

- **SWI Naive :** CPU like naive version.
- **SWI+SRP+LM :** Shift Register Pattern (FIFO queue) with local memory optimization to increase bandwith.
- **ND+Naive :** GPU like naive version (shared local memory).
- **ND+2CU :** Kernel replication of the ND+Naive version.
- **ND+LM+MP :** Last implemented optimization. Focused on a prefetching mechanism bound to the algorithm specificity. The goal is to access the pattern needed by a group of voxel in a single call.

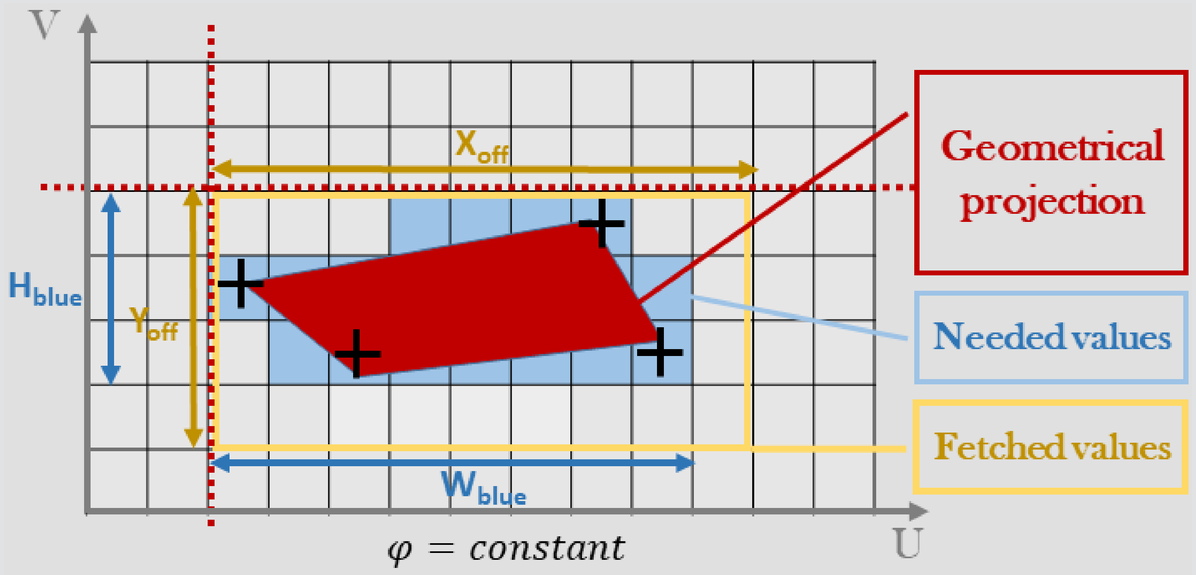
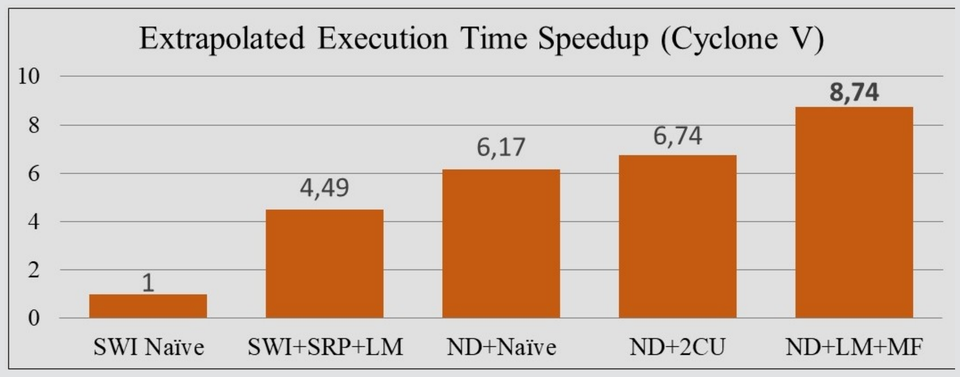


Figure 3: Sinogram memory fetching pattern optimization.

FPGA obtained speedup and GPU comparison



- The Altera Offline Compiler is effective in guaranteeing no kernel stall.
- Optimizing a SWI kernel lies with **optimizing memory handling** and data streaming effectiveness in order to increase kernel frequency.
- Optimizing a NDRange kernel coincide with **reducing the logical footprint** to allow more kernel replication within the same chip.

Device	Power (W)	Execution time (ms)	Energy (mWh)
Titan X Pascal	250	12	0.83
Jetson TK2	15	94	0.39
Intel Arria 10	2.27	991	0.63

Even with the extrapolated results on an Aria 10 FPGA, and with the obtained improvements over the Naive versions, the FPGA has a much longer execution time than the GPUs, mostly due to the back-projection algorithm being adapted to SIMD architectures.

Conclusion

1. With an overall **8.74 speedup** between naive and optimized kernels, there is room for more improvements closely related to memory handling.
2. On this application, FPGAs still fall short compared to GPUs, mostly due to the algorithm inadequacy for pipeline programming.

References

[1] Kavya Shagrithaya, Krzysztof Kepa, and Peter Athanas. Enabling Development of OpenCL Applications on FPGA platforms. *ASAP*, 2013.

[2] H. Lu, Cheng J., Han G., Li L., and Liang Z. A 3D distance-weighted Wiener filter for Poisson noise reduction in sinogram space for SPECT imaging. *Physics of Medical Imaging*, 2001.

Acknowledgments

We would like to thank Terasic, CNFM, and Intel for providing hardware and software used in this project, and Nicolas Gac for his supervision.