A new convolutions algorithm to leverage tensor cores

Mickael Seznec, Nicolas Gac, F. Orieux, A. Sashala Naik

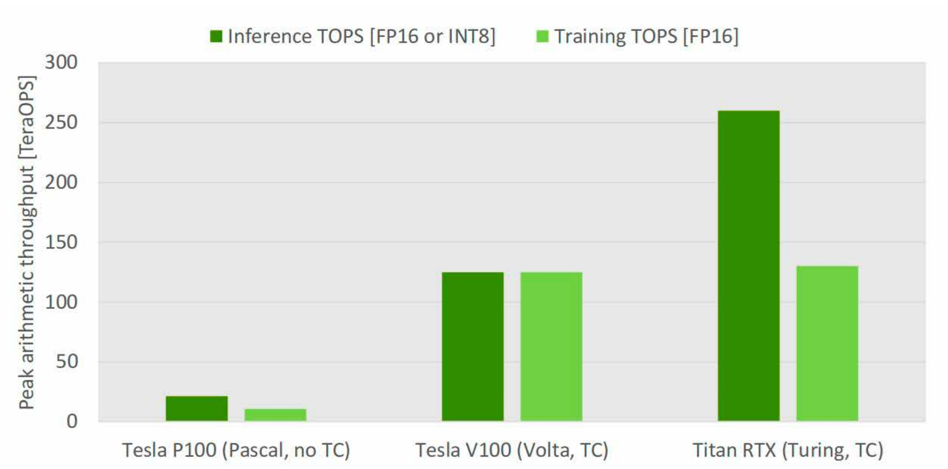
POSTER
P21960
CONTACT NAME
Mickaël Seznec: mickael.seznec@centralesupelec.fr

NVIDIA GTC GPU TECHNOLOGY CONFERENCE

A New Convolution Algorithm to Leverage Tensor Cores

Context

Tensor Cores promise 12X the throughput for matrix multiplication (GEMM)



- Tensor cores are shipped with Nvidia GPUs since Volta (2017).
- Dedicated hardware for matrix multiplication.
- Handle sub-FP32 precision

Convolutions in CNNs already leverage Tensor Cores...

- Convolution are not natively matrix multiplication
- The im2col algorithm is used to rearrange batched convolutions as a matrix multiplication
- Implementation available in cuDNN.

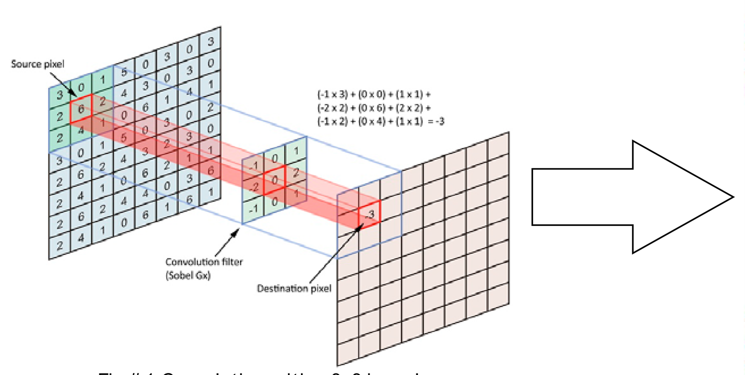


Fig # 1: Convolution with a 3x3 kernel

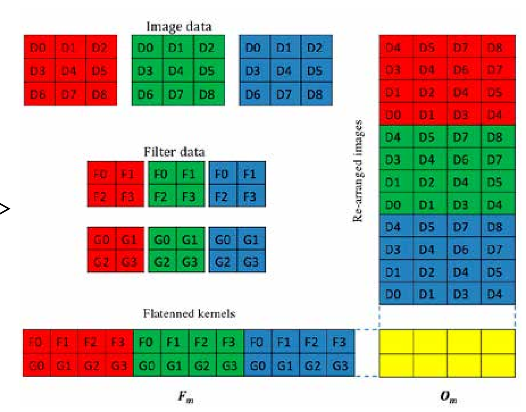
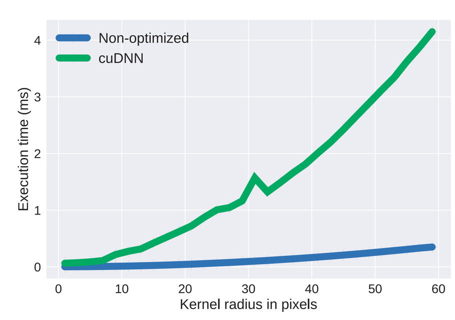


Fig # 2: Im2col rewrites convolutions a GEMM

...But poor performances with traditional workloads

- With a single image, single kernel, im2col becomes a vector / matrix multiplication
- Low artihmetic complexity = execution limited by bandwidth



Plot # 1: Execution time vs Kernel radius on RTX2080. Lower is better

Proposed algorithm

Definitions and first approach:

- Does not use the circulant matrix associated with the convolution
- Use the image to build sub-matrices directly with some overlapping
- Convolution kernel is transposed
- Collect final result with a sum on the diagonals
- Leads to a (K, K) x (K, H x W) matrix multiplication = better arithmetic complexity!

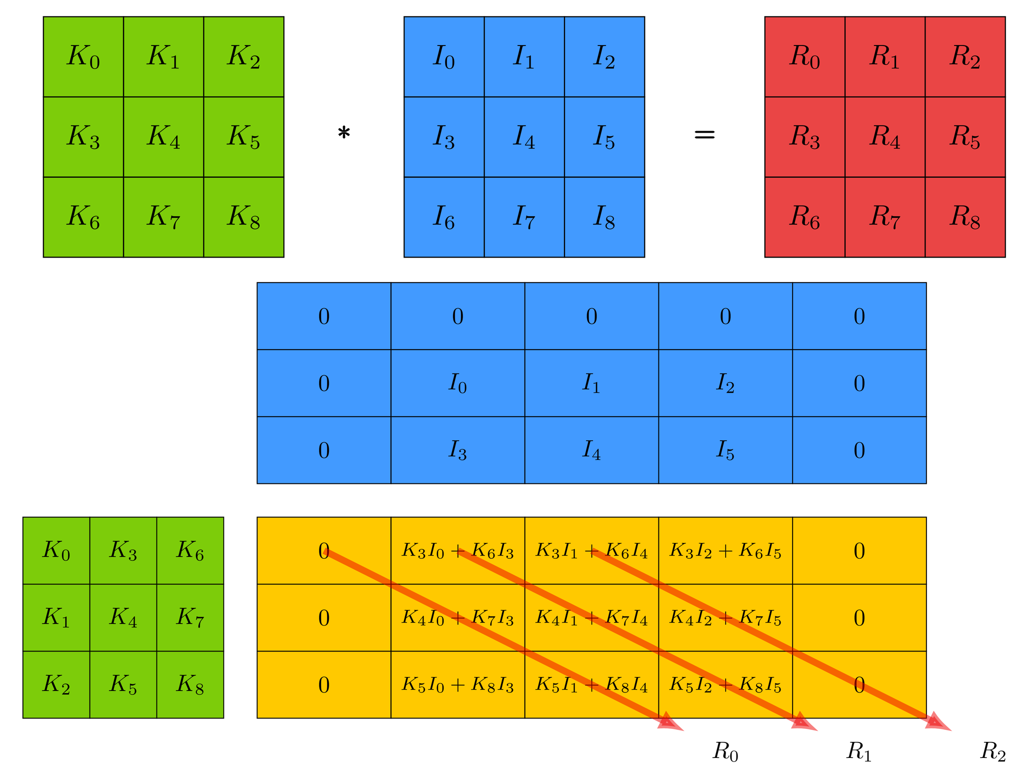
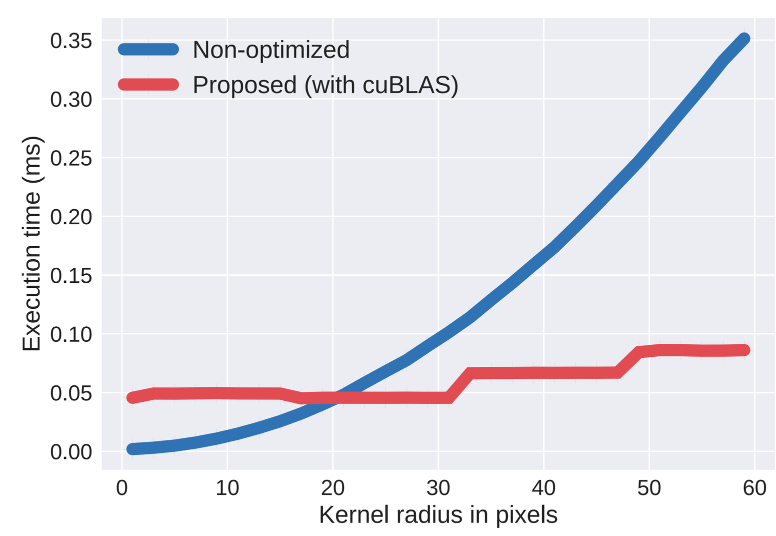


Fig # 3: our algorithm computes one line of the result with a GEMM + diagonal sum convolutions a GEMM



Plot # 2: Execution time vs Kernel radius on RTX2080. Lower is better

Initial implementation with cuBLAS:

- The input matrix is built explicitely but duplicating image lines -> takes time and memory
- But rely on highly optimized GEMMs with cuBLAS
- Comparaison of the new implementation with the simple version (the time to create the matrix is not counted)

Defining the matrix implictly:

- The matrix does not have to be created explicitly: CUDA blocks know which line to fetch from the image
- Allows image lines to be re-used in the same CUDA kernel, either from caches or shared memory
- Complexify a little the code of the algorithm

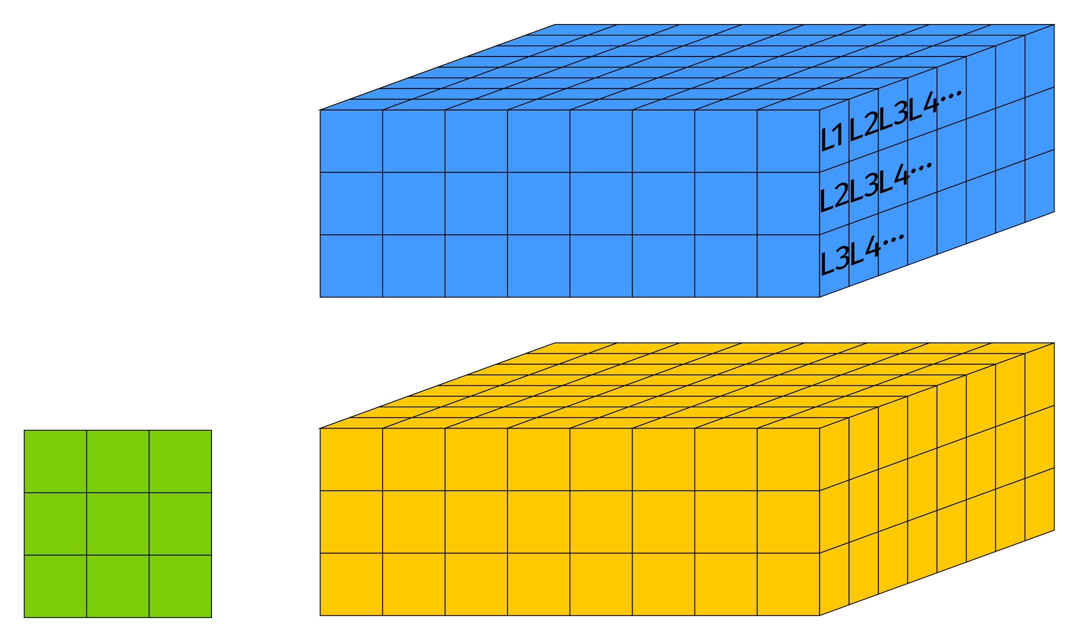


Fig # 4: we can view our algorithm as a GEMM broadcast to a 3D-tensor

Results

Using Nsight-Compute to assess Tensor Cores performance

- Nsight-Compute is available since CUDA 10.0 (2018)
- Is replacing nvvp and nvprof
- Can show Tensor Cores utilization
- We tried to reach the performance from the GEMM example in the CUDA samples

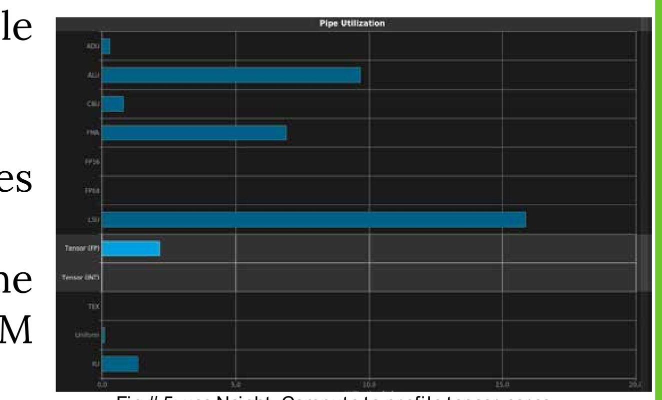
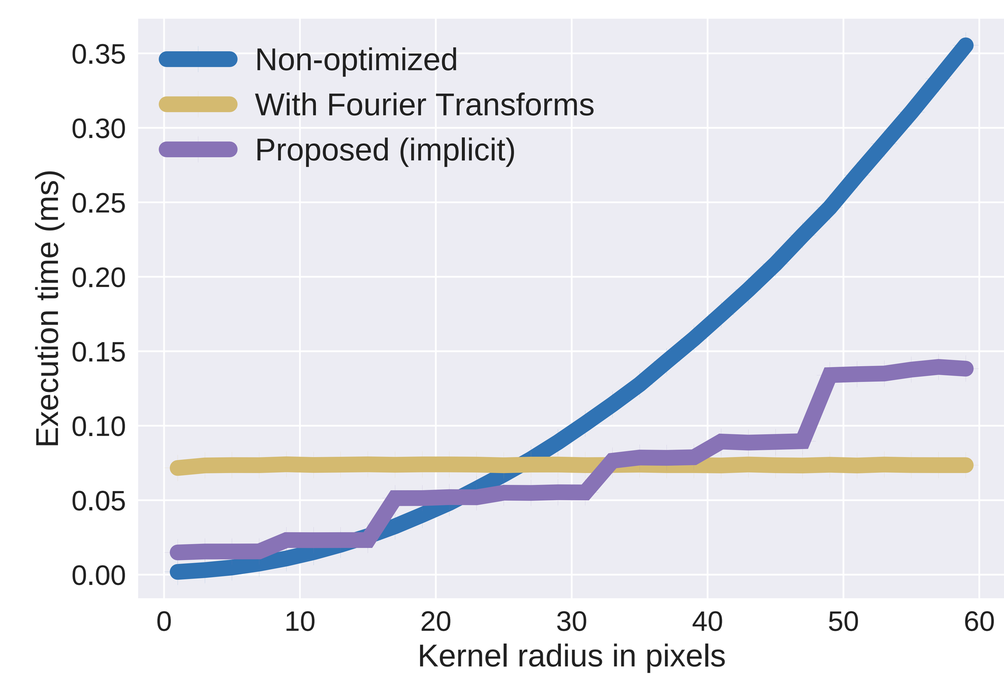


Fig # 5: use Nsight-Compute to profile tensor cores

First results:



Plot # 3: Execution time vs Kernel radius on RTX2080. Lower is better

- Real impact when the kernel is big
- On par with smaller kernel
- FFT is asymoptically better
- Still need some optimizations to outperform other libraries on a larger scale of kernels

References:

- Chetlur, S., Woolley, C., Vandermersch, P., Cohen, J., Tran, J., Catanzaro, B., & Shelhamer, E. (2014). cudnn: Efficient primitives for deep learning.
- Vasudevan, A., Anderson, A., & Gregg, D. (2017, July). Parallel multi channel convolution using general matrix multiplication.
- Seznec, M., Gac, N., Ferrari, A., & Orieux, F. (2018, October). A Study on Convolution using Half-Precision Floating-Point Numbers on GPU for Radio Astronomy Deconvolution. (IEEE SiPS)

We designed a new algorithm for **2D-convolutions** that uses **tensor cores.** Contrary to cuDNN implementation, it **does not rely on batched kernels** for efficiency. Results are **already better than current libraries in some contexts** and could be further optimized.

Scan me for more information & digital version

Mickaël Seznec[1,2] (mickael.seznec@centralesupelec.fr), Nicolas Gac[1], Francois Orieux[1], Alvin Sashala Naik[2]
[1]L2S, CentraleSupelec, Gif-Sur-Yvette, France [2]Thales Research and Technology, Palaiseau, France