



HAL
open science

A new convolutions algorithm to leverage tensor cores

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

► **To cite this version:**

Mickael Seznec, Nicolas Gac, F. Orieux, A. Sashala Naik. A new convolutions algorithm to leverage tensor cores. GPU Technology Conference (GTC), May 2020, Silicon Valley, United States. hal-02605077

HAL Id: hal-02605077

<https://hal.science/hal-02605077>

Submitted on 16 May 2020

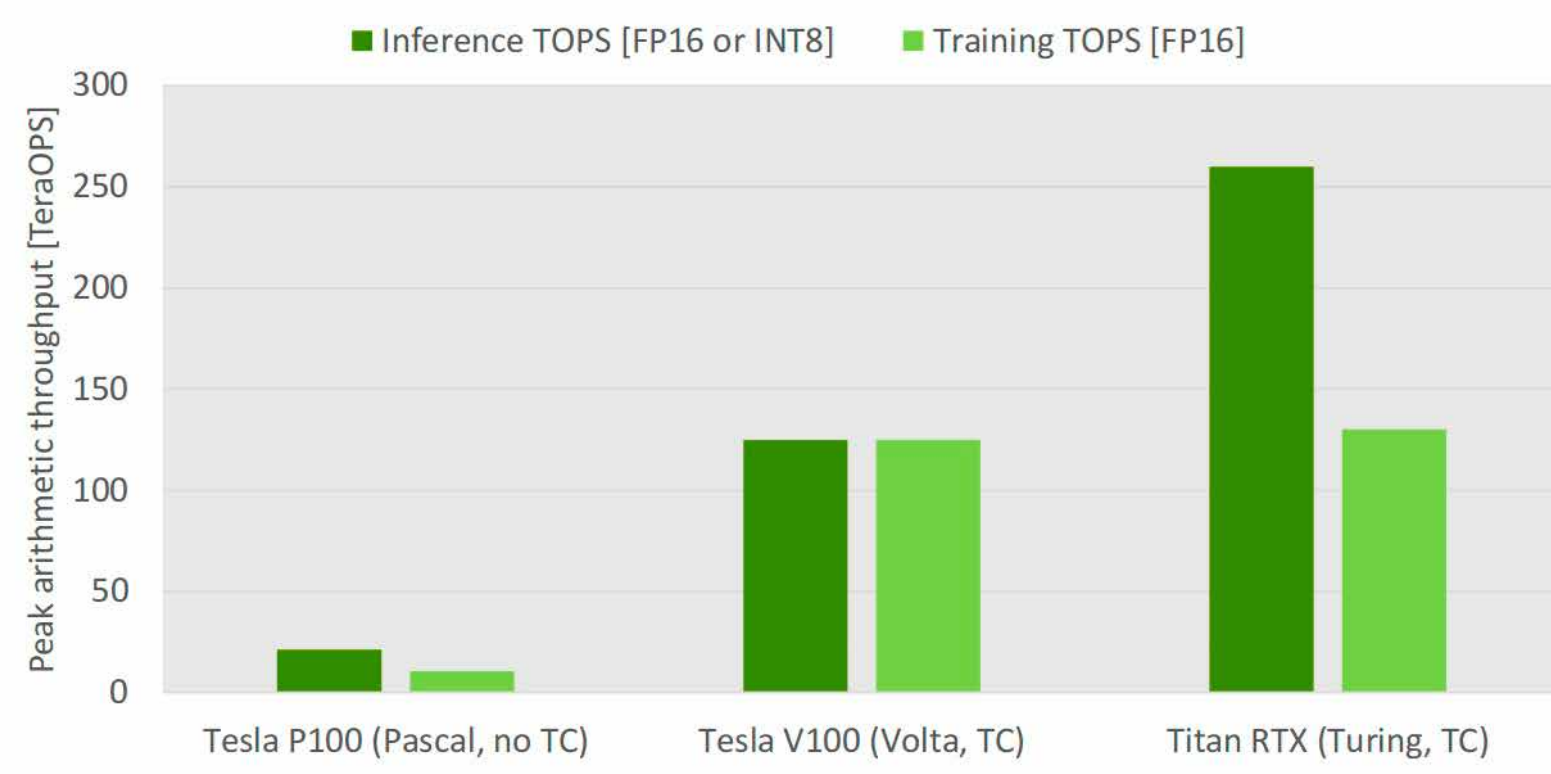
HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A New Convolution Algorithm to Leverage Tensor Cores

Context

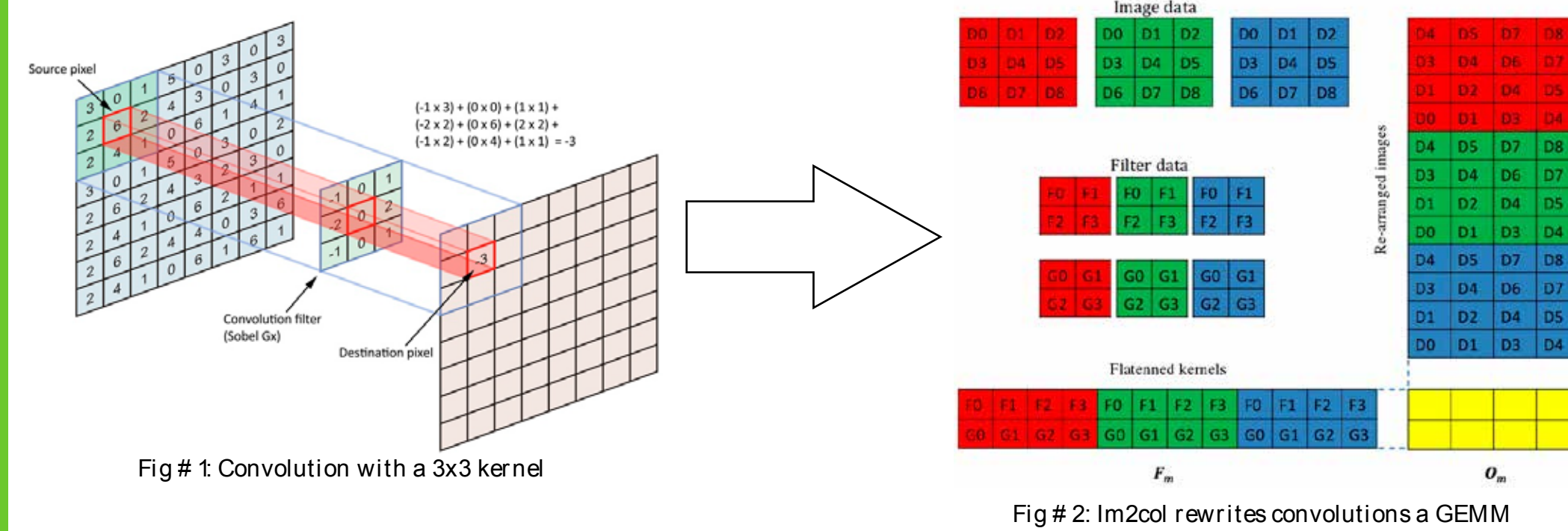
Tensor Cores promise 12X the throughput for matrix multiplication (GEMM)



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

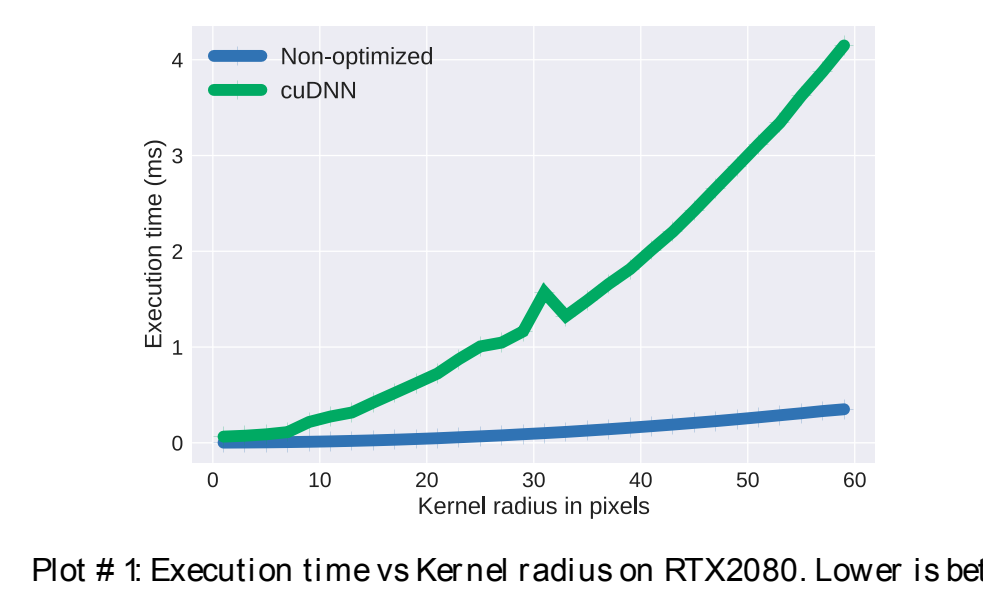
Convolutions in CNNs already leverage Tensor Cores...

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



...But poor performances with traditional workloads

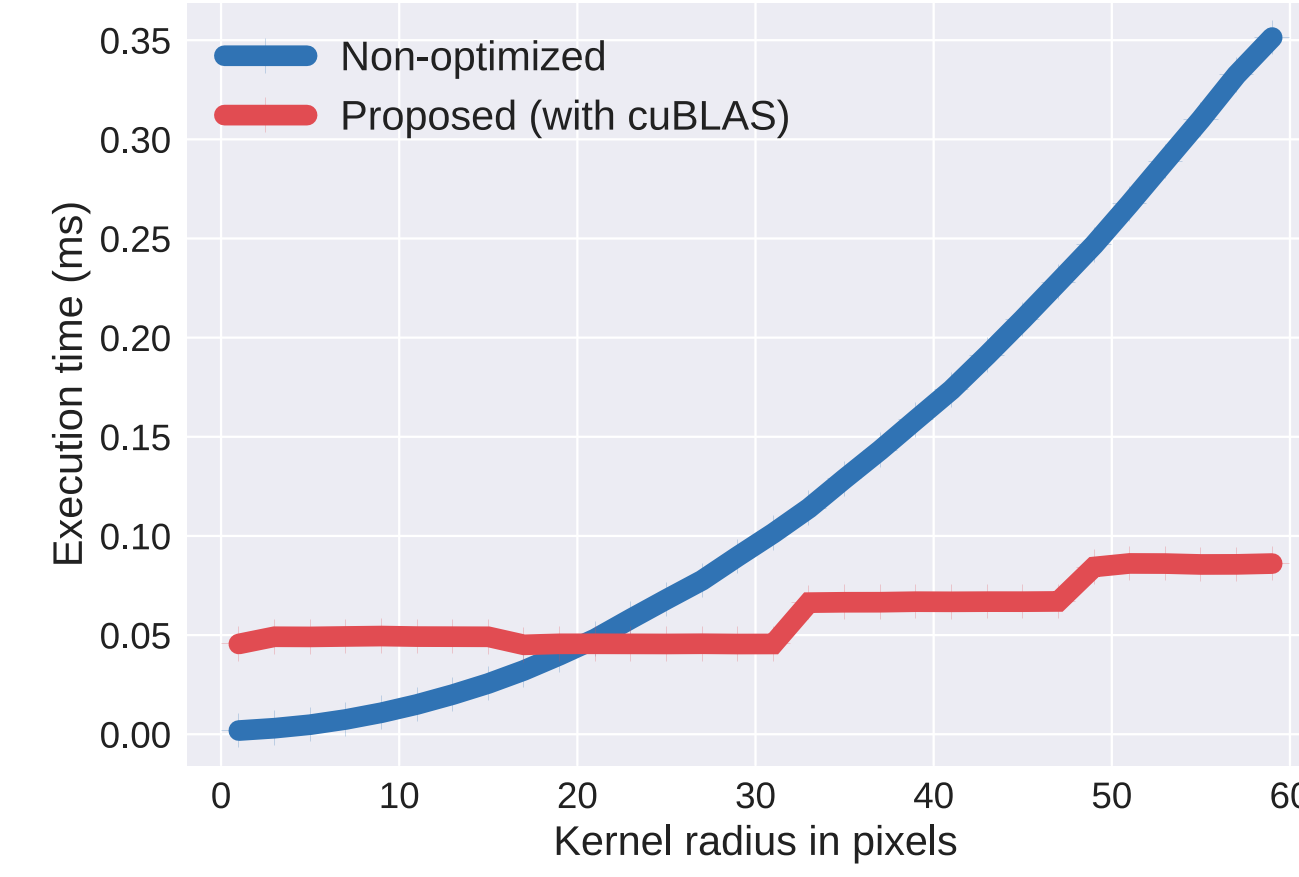
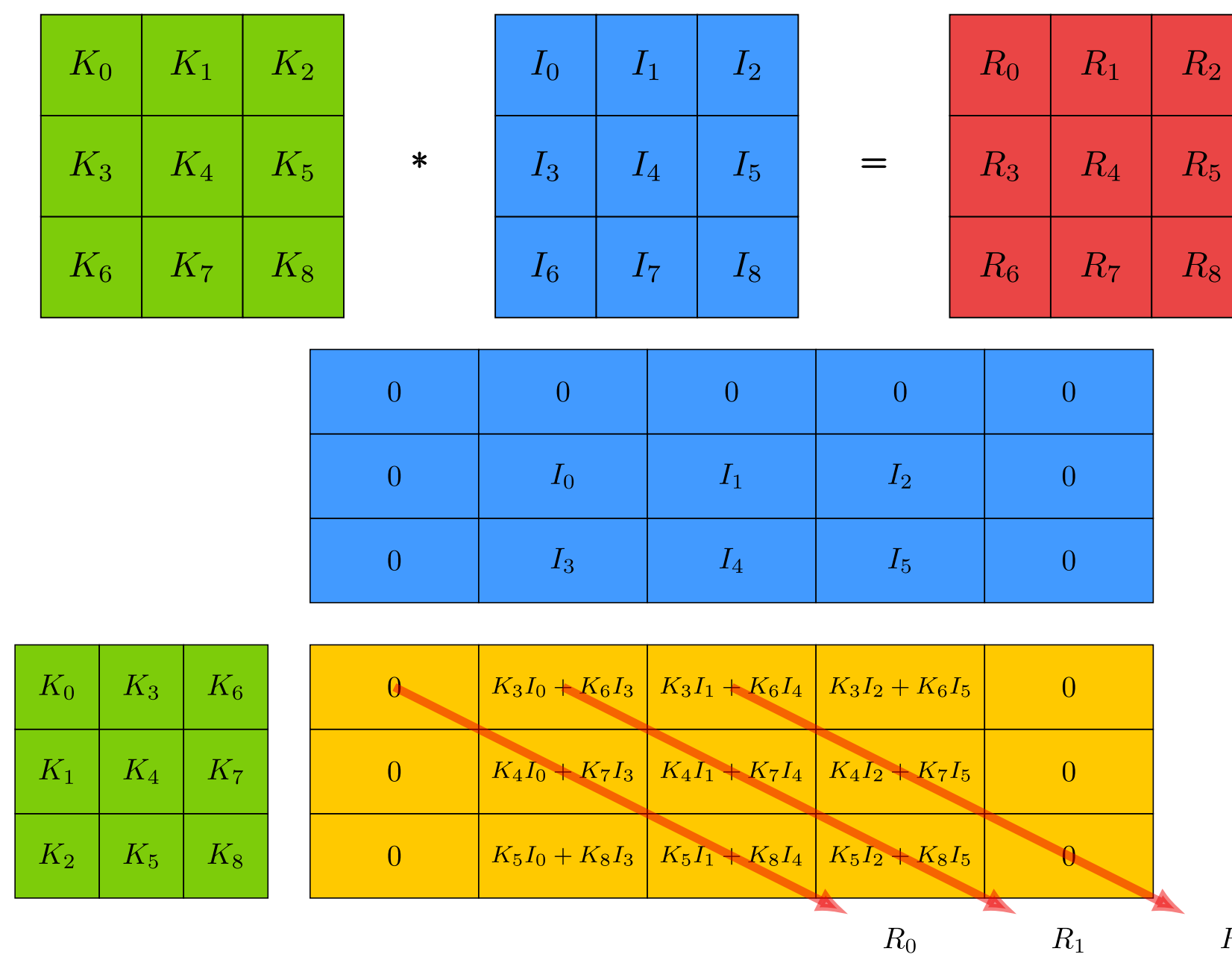
- With a single image, single kernel, im2col becomes a vector / matrix multiplication
- Low arithmetic complexity = execution limited by bandwidth



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!

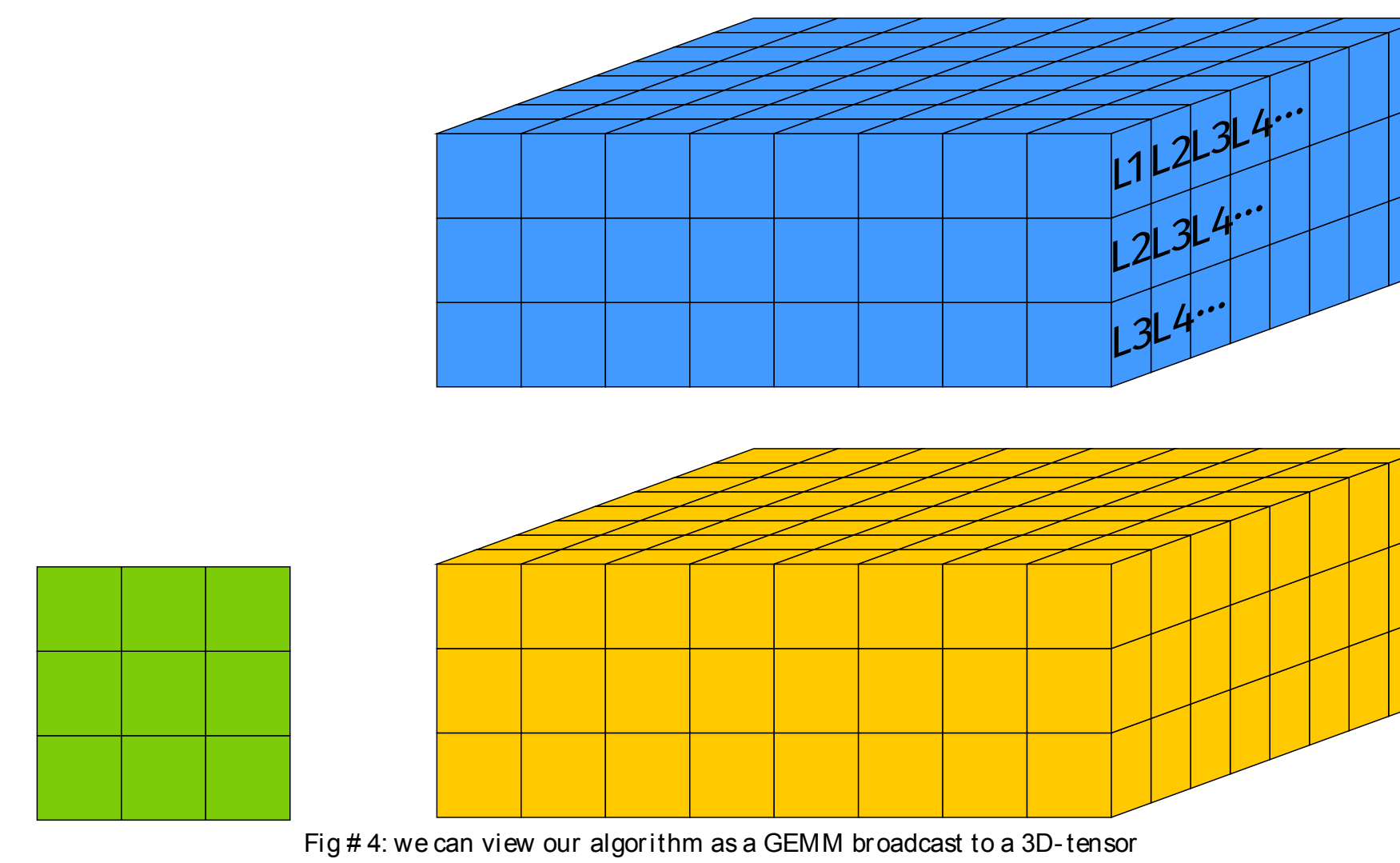


Initial implementation with cuBLAS:

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

Defining the matrix implicitly:

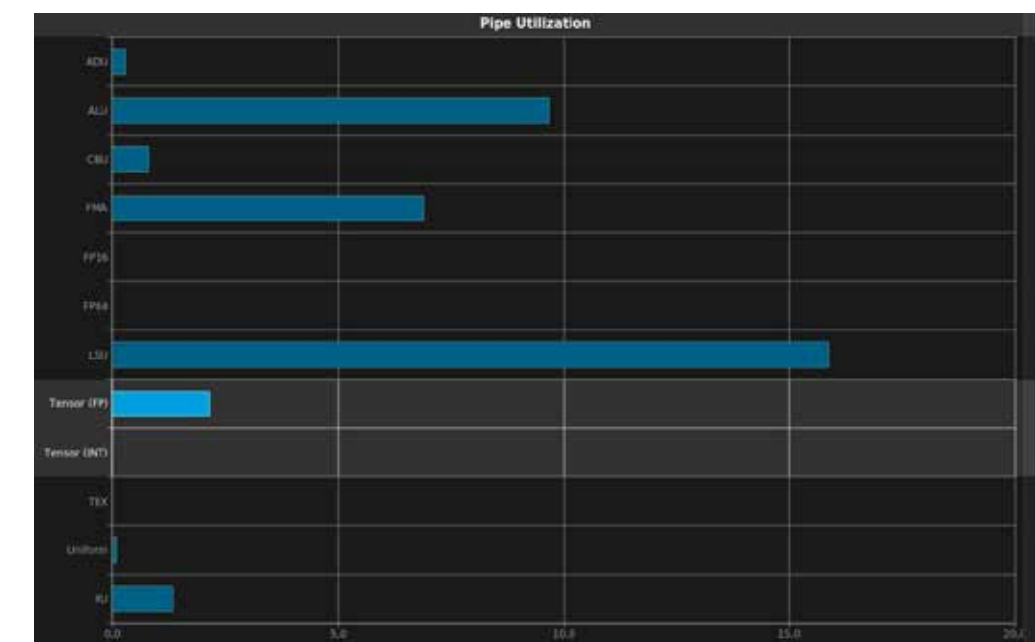
- The matrix does not have to be created explicitly: CUDA blocks know which line to fetch from the image
- Allows image lines to be re-used in the same CUDA kernel, either from caches or shared memory
- Complexify a little the code of the algorithm



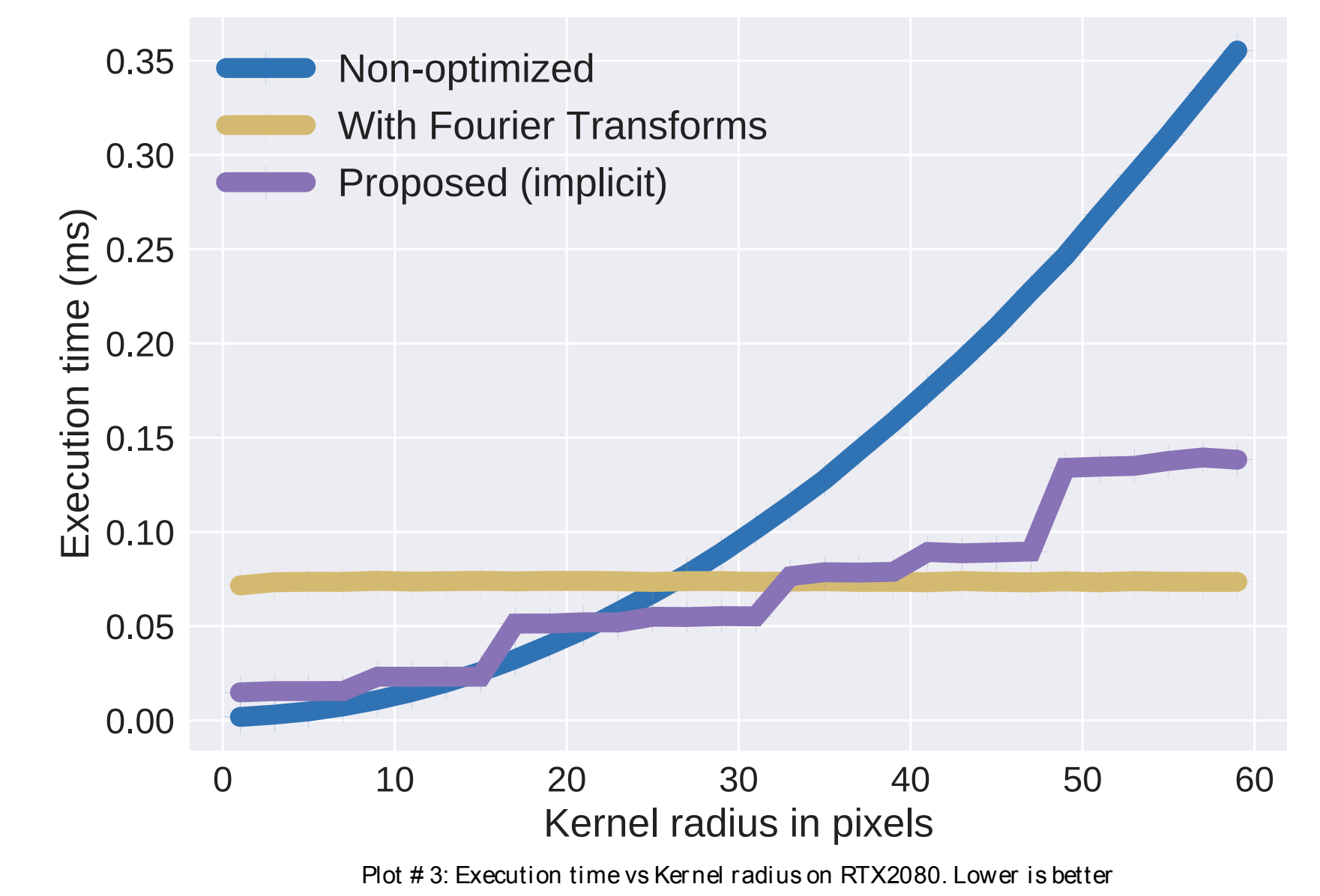
Results

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



First results:



- Real impact when the kernel is big
- On par with smaller kernel
- FFT is asymptotically 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., & Orioux, 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