

Research Project

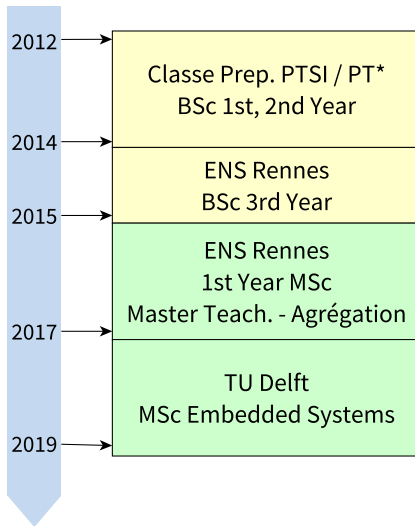
Acceleration of non-rigid image registration with Tensor Cores

Jonathan LEVY

June 17, 2019

About me

- Jonathan LEVY
- MSc student in Computer Science
- Engineering background



Cursus Summary

Since September 2019:

GASAL2 : GPU-accelerated library for DNA alignment

Languages C/C++ and CUDA

Algorithm Smith-Waterman - optimal alignment for short pair

Goal Speed-up the *Burrough-Wheeler Aligner*, "BWA" by 1.33x

<https://github.com/j-levy/GASAL2>

<https://github.com/j-levy/bwa-gasal2> ← private repository

<https://jlevy.weblog.tudelft.nl> ← weekly logs

Acceleration of non-rigid image registration with Tensor Cores

- Image registration: aligning a *floating* image with a *reference*.
- *Non-rigid*: various deformations allowed
- Use Next-gen GPU for acceleration

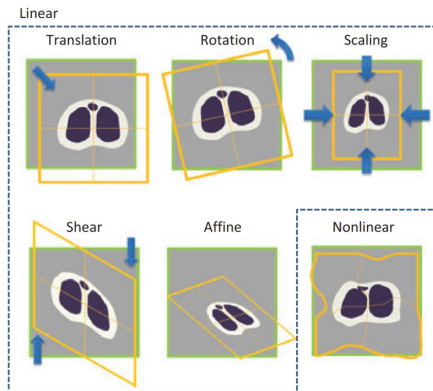


Figure 1: Different types of deformation.

Acceleration with Tensor Cores

Recent NVIDIA GPUs
(Volta Architecture)

- Refined scheduler
- New memory scheme
- Tensor Cores

Tensor Cores:

WHAT Matrix-matrix multiplication

HOW Mixed precision (precision loss)

WHY Originally, deep learning

$$D = \begin{pmatrix} A_{0,0} & A_{0,1} & A_{0,2} & A_{0,3} \\ A_{1,0} & A_{1,1} & A_{1,2} & A_{1,3} \\ A_{2,0} & A_{2,1} & A_{2,2} & A_{2,3} \\ A_{3,0} & A_{3,1} & A_{3,2} & A_{3,3} \end{pmatrix} \begin{pmatrix} B_{0,0} & B_{0,1} & B_{0,2} & B_{0,3} \\ B_{1,0} & B_{1,1} & B_{1,2} & B_{1,3} \\ B_{2,0} & B_{2,1} & B_{2,2} & B_{2,3} \\ B_{3,0} & B_{3,1} & B_{3,2} & B_{3,3} \end{pmatrix} + \begin{pmatrix} C_{0,0} & C_{0,1} & C_{0,2} & C_{0,3} \\ C_{1,0} & C_{1,1} & C_{1,2} & C_{1,3} \\ C_{2,0} & C_{2,1} & C_{2,2} & C_{2,3} \\ C_{3,0} & C_{3,1} & C_{3,2} & C_{3,3} \end{pmatrix}$$

FP16 or FP32 FP16 FP16 FP16 or FP32

Figure 2: Operation done by a Tensor Core

Could be used to calculate:

- B-Splines (quantify smoothness)
- Entropy (quantify similarity)

And other various modern optimizations

Integrate in existing work:

- ① Accelerate B-Splines calculation using tensor cores
- ② Accelerate entropy calculation with tensor cores too
- ③ Quantify precision loss
- ④ Allow for precision refining if needed
- ⑤ Send results for rendering (visual output)

Why Japan?

- Leading role in HPC
- Culture,