

Research Project

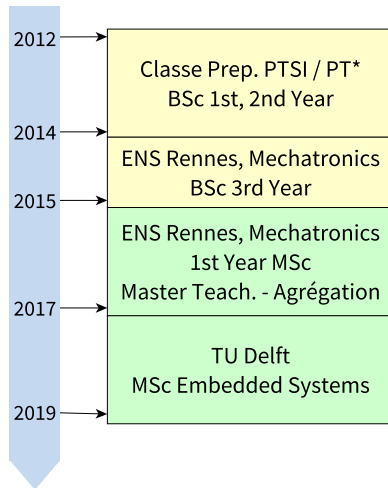
Acceleration of non-rigid image registration with Tensor Cores

Jonathan LEVY

June 18, 2019

About me:

- Jonathan LEVY
- MSc student in Embedded Systems, TU Delft (Netherlands)
- Multiple majors & countries



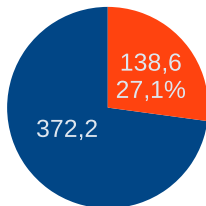
GASAL2 : GPU-accelerated library for DNA alignment

Languages C/C++ and CUDA

Algorithm Smith-Waterman

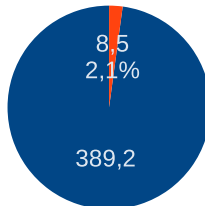
Goal Speed-up the *Burrough-Wheeler Aligner*, "BWA" by 1.33x

Fraction of time for alignment in
BWA (12 threads)



■ Rest (s) ■ Alignment (s)

Fraction of time for alignment in BWA-
GASAL2 (with hidden time, 12 threads)



■ Rest (s) ■ Alignment (s)

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

Weekly logs → <https://jlevy.weblog.tudelft.nl>

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

Acceleration of non-rigid image registration with Tensor Cores

- Image registration: aligning 2 images
- *Non-rigid*: various deformations allowed
- Use next-gen GPUs for acceleration
- **Goal: get closer to real-time (currently: seconds) for surgery**

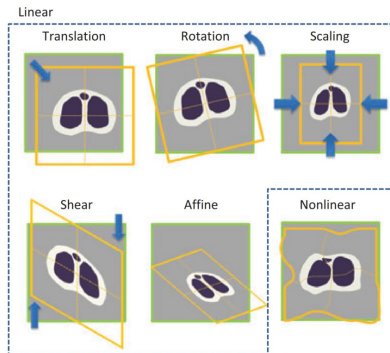


Figure 1: Different types of deformation.

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 (image deformation, quantify smoothness)
- Entropy (quantify similarity)

And other various modern optimizations

Provide a library for accelerated calculation:

- ① Accelerate entropy (NMI) with tensor cores
 - ② Accelerated B-Splines using tensor cores too
 - ③ Quantify precision loss
 - ④ Send results for rendering (visual output)
 - ⇒ Generic functions
 - ⇒ Reusable
-

Challenges:

Sufficient speedup? Integration in another software? Precision loss?

Why Japan?

ENS Rennes: French state school teachers and researchers
Yet: *few incentives to go abroad!!*

- Entice younger students to go abroad
- Foster global research, in the EU and outside
- Personal interest ↗

Contacted 2 laboratories:



Pr. Rio YOKOTA

Tokyo Tech

Pr. Fumihiko INO



OSAKA UNIVERSITY

Helped defining the project.