Research Project

Acceleration of non-rigid image registration with Tensor Cores

Jonathan LEVY

June 17, 2019

Outline

Cursus

About me

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

Cursus Summary

- *Classe Préparatoire PTSI/PT**
- Ecole Normale Supérieure de Rennes (BSc, Master in Teaching)
- *Agrégation* in Engineering, CS track
- MSc Embedded Systems, TU Delft

Current work

Since September 2019:
GASAL2 : GPU-accelerated library for DNA alignment

When	First as Extra Project, then MSc Thesis
Languages	C/C++ and CUDA
Algorithm	Smith-Waterman - optimal alignment for short pair

Goal: integrate in the *Burrough-Wheeler Aligner, "BWA"*

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 GPU for parallel calculation

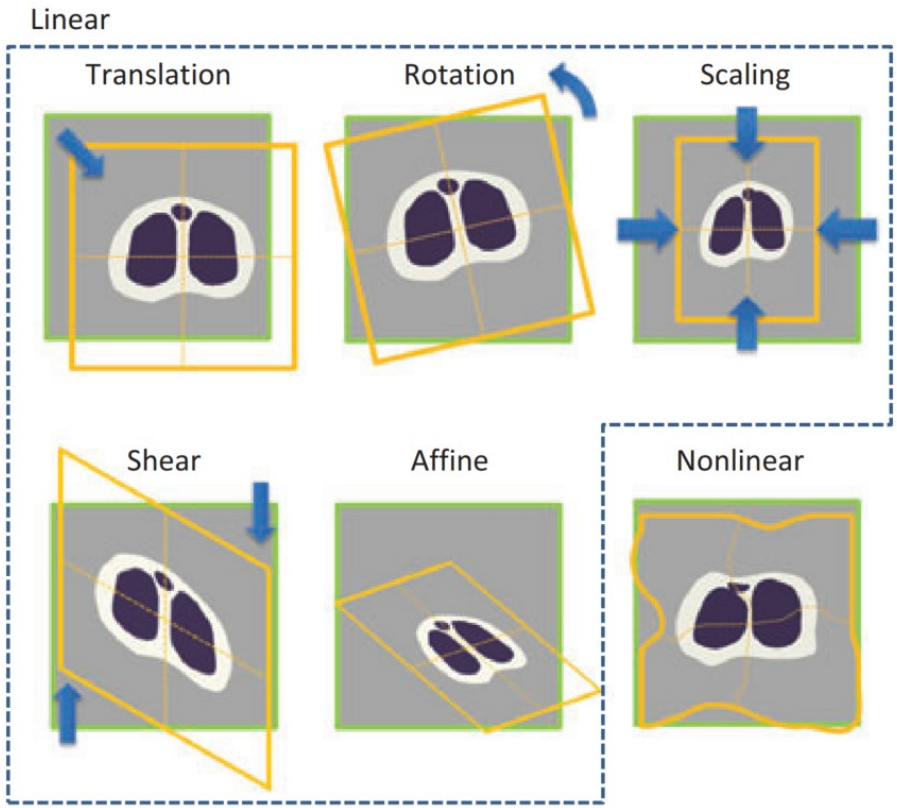


Figure 1: Different types of deformation.

The Volta Architecture

- NVIDIA GPUs' architecture (2017)
- Several changes:
  - HBM2 memory
  - Parallel FP/Integer calculation
  - Tensor Cores

Figure 2: The full GV100 architecture

The Volta Architecture

- NVIDIA GPUs' architecture (2017)
- Several changes:
  - HBM2 memory
  - Parallel FP/Integer calculation
  - Tensor Cores

Figure 2: Volta Streaming Multiprocessor (80 units per GV100)

The Volta Architecture

- NVIDIA GPUs' architecture (2017)
- Several changes:
  - HBM2 memory
  - Parallel FP/Integer calculation
  - Tensor Cores

Figure 2: Volta Processing Block (4 units per SM)

Tensor Cores

WHAT Matrix-matrix multiplication

HOW Mixed precision (precision loss)

WHY Deep Learning

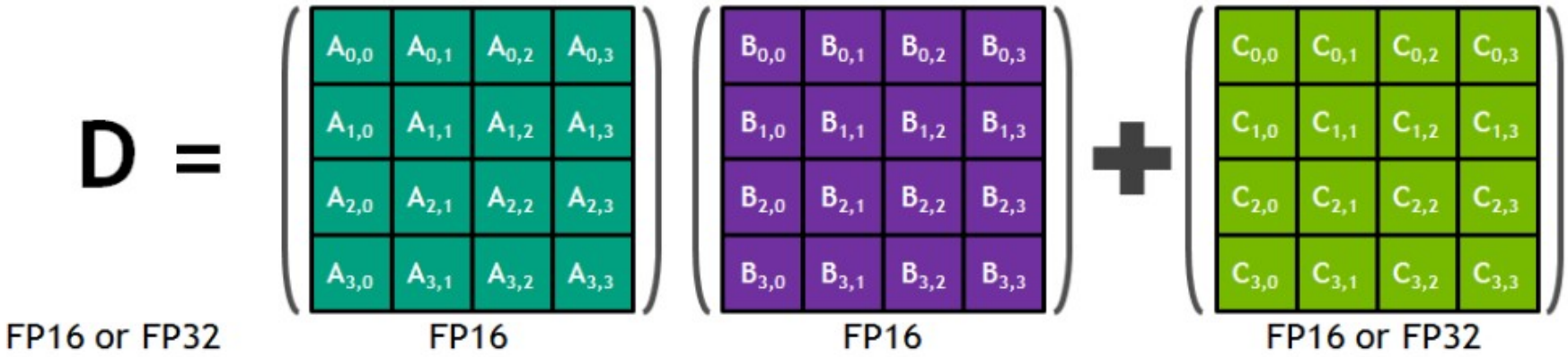


Figure 3: Operation done by a Tensor Core

Registration steps

Minimize a cost function

$$C\left(\theta, \Phi\right) = -C_{similarity}\left(I\left(t_{\mathrm{o}}\right), T\left(I\left(t\right)\right)\right) + \lambda C_{smooth}\left(T\right) \tag{1}$$

1. Calculate gradient $\nabla C = \frac{\partial C(\theta, \phi_l)}{\partial \phi_l}$
2. While $||\nabla C|| > \varepsilon$:
   1. Update control points: $\phi = phi + \mu \frac{\nabla C}{||\nabla C||}$
   2. Make a tighter net (higher resolution)

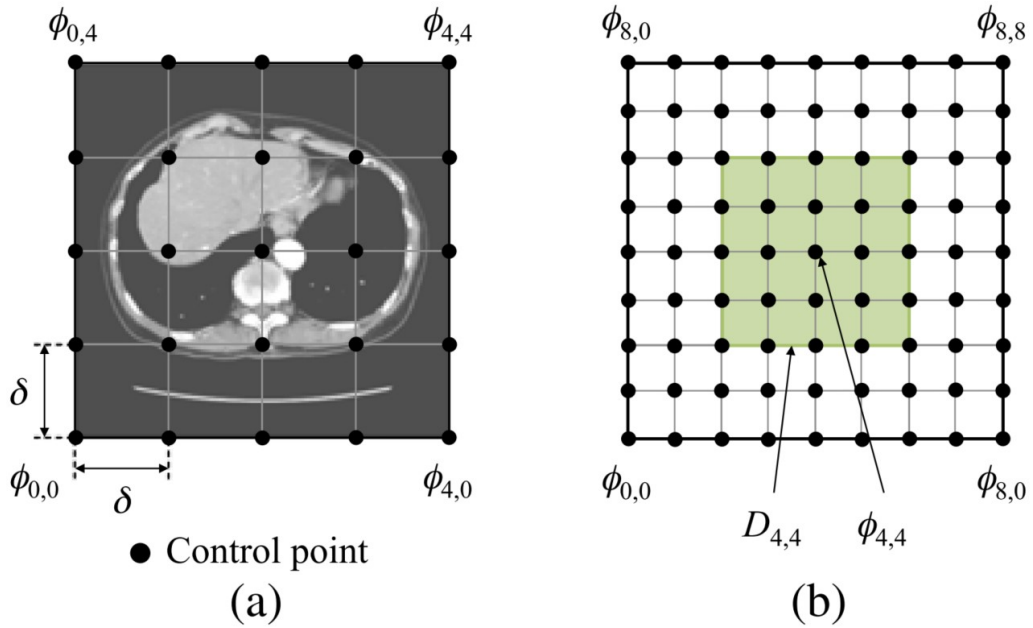
B-Splines model

GOAL Find optimal transformation $T : (x, y, z) \longmapsto (x', y', z')$

ALGO Spline-based Free-Form Deformation (FFD) : 3D deformation model using net of points $\phi_{x,y,z}$

$$T(x, y, z) = \sum_{l=0}^{3} \sum_{m=0}^{3} \sum_{n=0}^{3} B_l(u) B_m(v) B_n(w) \phi_{i+l,j+m,k+n} \quad (2)$$

Tensor product $\Longrightarrow$
Calculation by Tensor Cores possible
Each point affects is 4 direct neighbours



Similarity score and Entropy

$C_{similarity}$ relies on histograms and entropy calculation.
A formula for entropy $H(X)$:

$$H(X) = -\sum_{i=1}^{n} P_i * log_2(P_i) \tag{3}$$

with:

- n the number of different values for pixels,
- P_i probablity distribution of the value i (values of histogram)

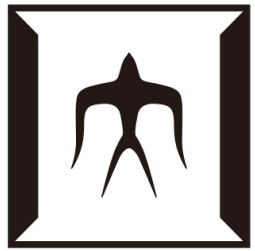
Sum of products : feasible with matrix-matrix multiplication
$\Longrightarrow$ Doable by Tensor cores.

Work proposal

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

Tokyo Institute of Technology

- Professor Rio YOKOTA
- Yokota lab: member of the Global Scientific Information and Computing Center *(GSIC)*
- HPC with CUDA



Tokyo Tech

Osaka University

Contacted supervisor: Professor Fumihiko INO

日本語のスライド

日本語のスライド 日本語のスライド