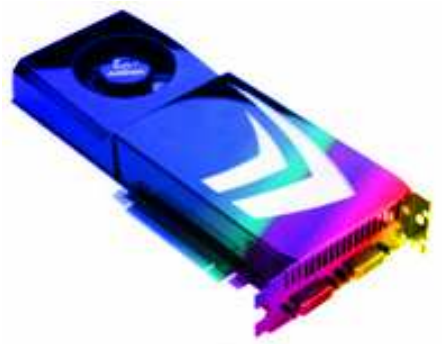
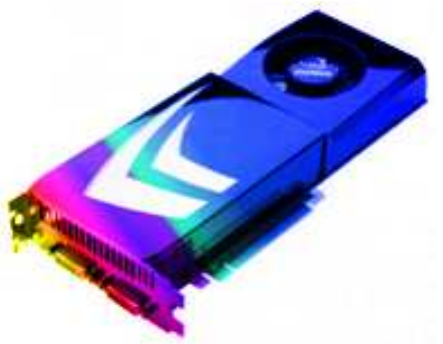
Acceleration through GPGPUs & FPGAs



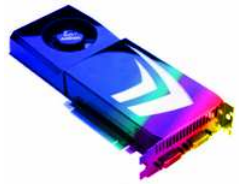
Gabriel Caffarena

Laboratorio de Bioingeniería
Universidad San Pablo CEU
Madrid - SPAIN

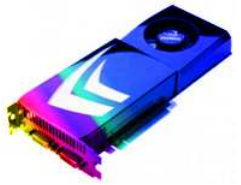


CEU
Universidad San Pablo

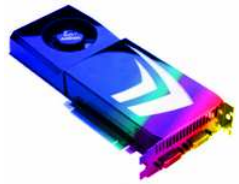
bio Engineering



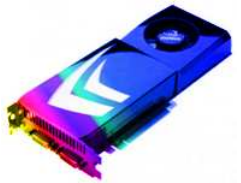
Agenda



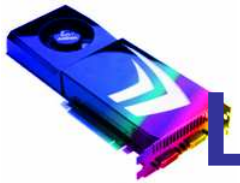
- Laboratory of Bioengineering
- HPC, GPUs and FPGAs
- GPGPU research
- FPGA research
- Acknowledgment



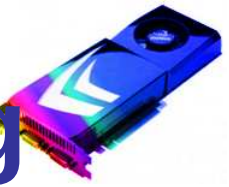
Agenda



- Laboratory of Bioengineering
- HPC, GPUs and FPGAs
- GPGPU research
- FPGA research
- Acknowledgement

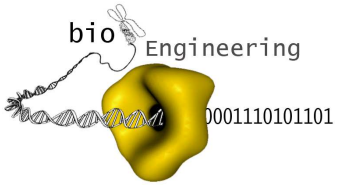


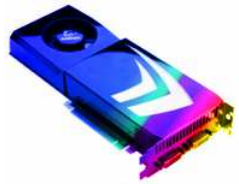
Laboratory of Bioengineering



The Bioengineering Laboratory from University CEU San Pablo The laboratory aims at solving problems coming from the Life Sciences and Medicine using Mathematical, Engineering and Computer Science techniques.

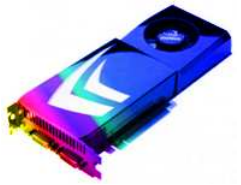
- **Image processing**
  - **Electron Microscopy (molecules and cells), PET, etc.**
- **Pattern recognition**
  - **ICU vital signs monitorization, etc.**
- **HW design and acceleration**
  - **FPGA/GPGPU acceleration**
  - **HLS and Fixed-Point optimization**



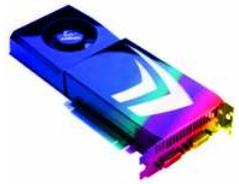


Laboratory of Bioengineering

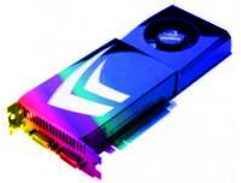
HW Acceleration



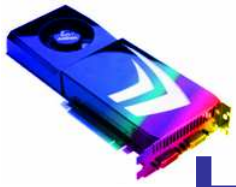
- **Use of GPUs and FPGAs to accelerate Scientific Applications**
- **Currently developing accelerated version of Electron Tomography software (XMIPP)**
- **GPU/FPGA library of main image processing functions**
- **Power analysis of GPU and FPGA technologies**
- **Development of arithmetic cores for FPGAs (fixed-point and floating-point)**

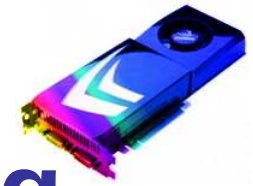


Agenda



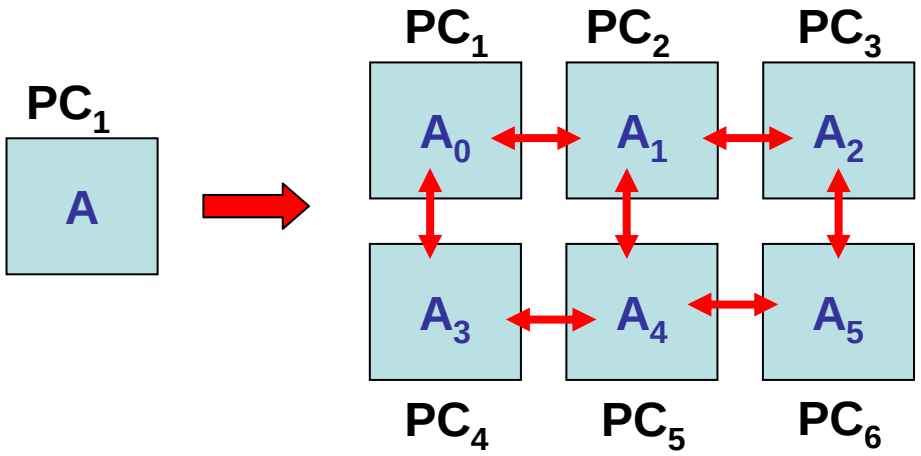
- Laboratory of Bioengineering
- HPC, GPUs and FPGAs
- GPGPU research
- FPGA research
- Acknowledgement





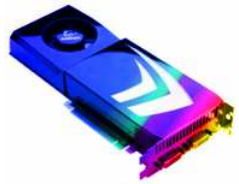
High-Performance Computing

- **HPC**: High-Performance Computing
- Use of many processors to execute a complex task in parallel: **PC cluster**

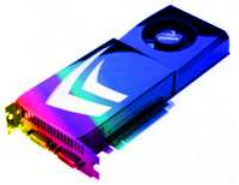


Is it 6 times faster?

What about power consumption?



Hardware Acceleration



- **CPU**

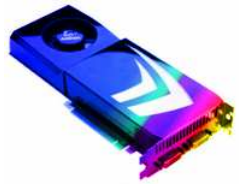
  Oriented to general-purpose processing (programmable)

- **FPGA**

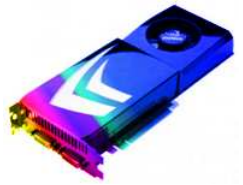
  Custom processing at very high speed (configurable HW)

- **GPGPU**

  Hundreds of processors executing the same task in parallel (programmable)



Hardware Acceleration



PC Cluster

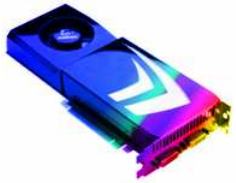
Pros

- **Software** development:
  - **MPI**, multi-threading, etc.
  - “Short” development times (SW compilation)

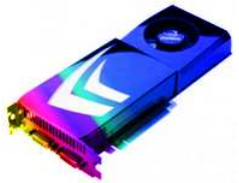
Cons

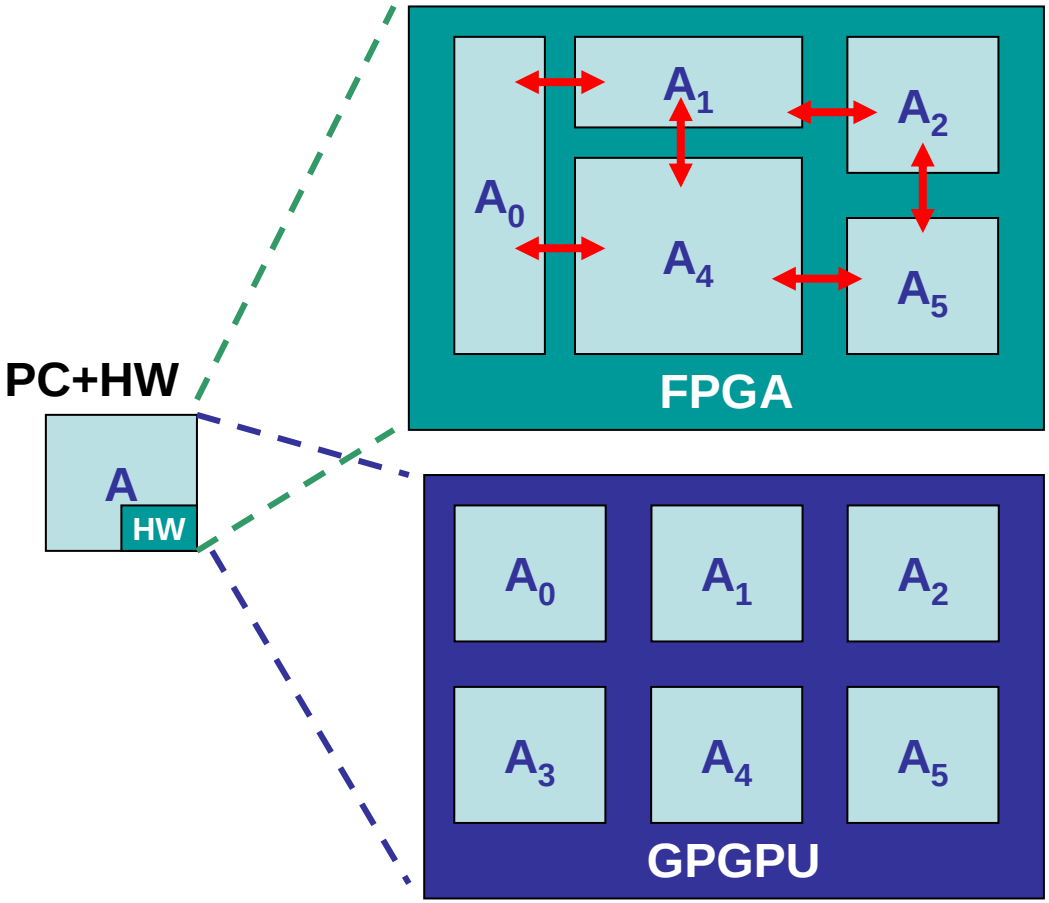
- Power consumption, heating
- Limited inter-node communicationlimitada

Parallel programming through MPI or multi-threading **is not an easy task**.



Hardware Acceleration





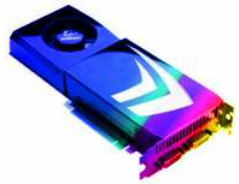
Can 1 PC + HW have a similar performance to a cluster?

Can it be more powerful?

Can we build a PC+HW cluster?

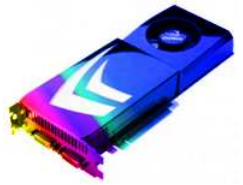
Power consumption?

Can we combine FPGAs and GPGPUs?



HPC, FPGAs & GPUs

Hardware Acceleration



HW-accelerated PC

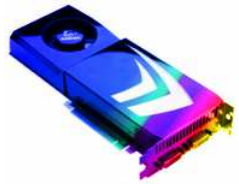
Pros

- Performance equal to or better than a cluster's
- Reduced power consumption
- Low cost

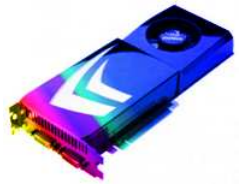
Pros

- Longer software development times
- Memory size limitation (4 GB, approx.)

Data dependencies determines if an application/algorithm is suitable for parallelization through FPGA or GPGPU and … PC clusters.



Amdahl's law

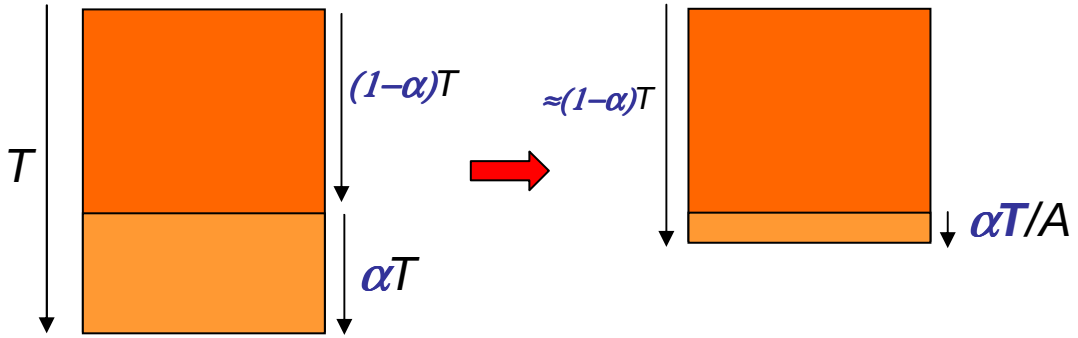


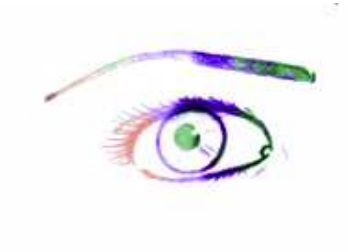
***T* = Original computing time**

α = fraction of *T* that is accelerated

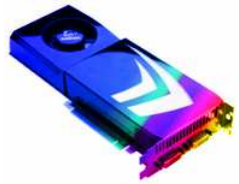
***A* = Acceleration with respect to αT**

$$A_{TOTAL} = \frac{1}{(1-\alpha)+\frac{\alpha}{A}} < \frac{1}{(1-\alpha)}$$

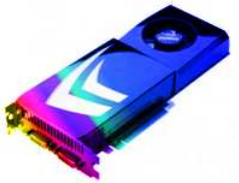




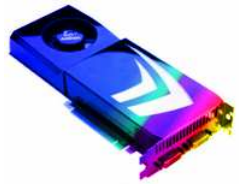
Goal: Try to reach values of α close to 1.



Agenda

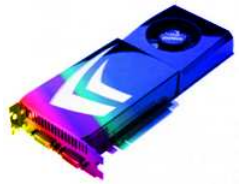


- Laboratory of Bioengineering
- HPC, GPUs and FPGAs
- GPGPU research
- FPGA research

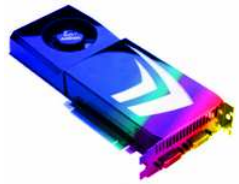


GPU Research

XMIPP

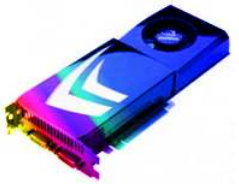


- **X-Window-based Microscopy Image Processing Package**
- **Developed at Spanish Center of Biotechnology (CNB-CSIC)**
- **European reference for Electron Microscopy**
  - Since 2008:
    - 800 users
    - 1600 downloads
- **GPGPU-accelerated version of main applications**
- http://xmipp.cnb.csic.es/

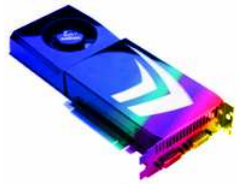


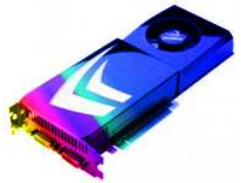
GPU Research

Dealing with legacy code



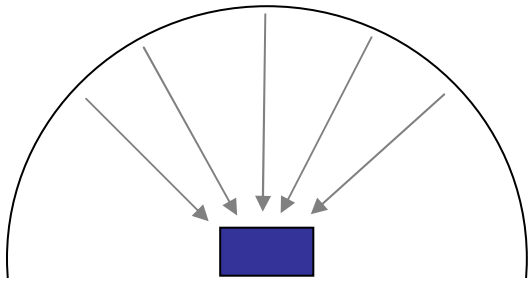
- **Everchanging code**
  - **Accelerate release version**
    - **Concerns about obsolescence**
  - **Accelerate common library**
    - **It might not be critical**
- **Developers against code flow changes**
  - **Code must be rewritten thinking on accelerator**
    - **Not flexible**
- **C++**
  - **Where is the code?**

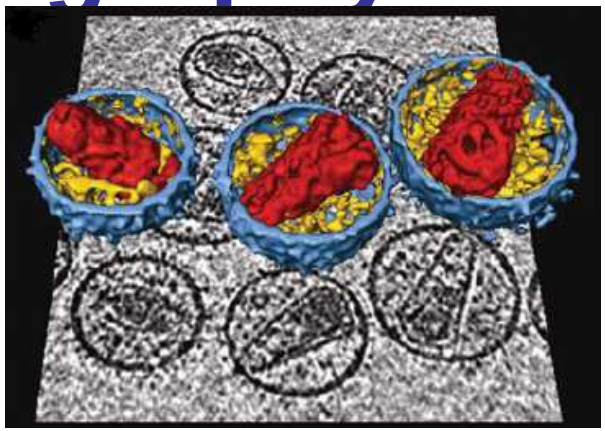




GPU Research

Electron Tomography





- Goal

  Extract 3D structure of macromolecular objects (cells, cell organelles, etc.)

- Input

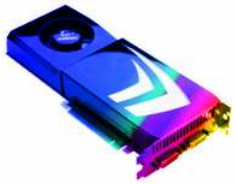
  Hundreds of images taken from different known angles

  (2048x2048-pixel B&W images)

- Problem

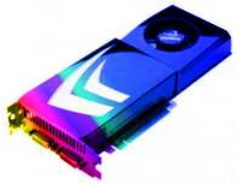
  Angle deviation due to mechanical problems

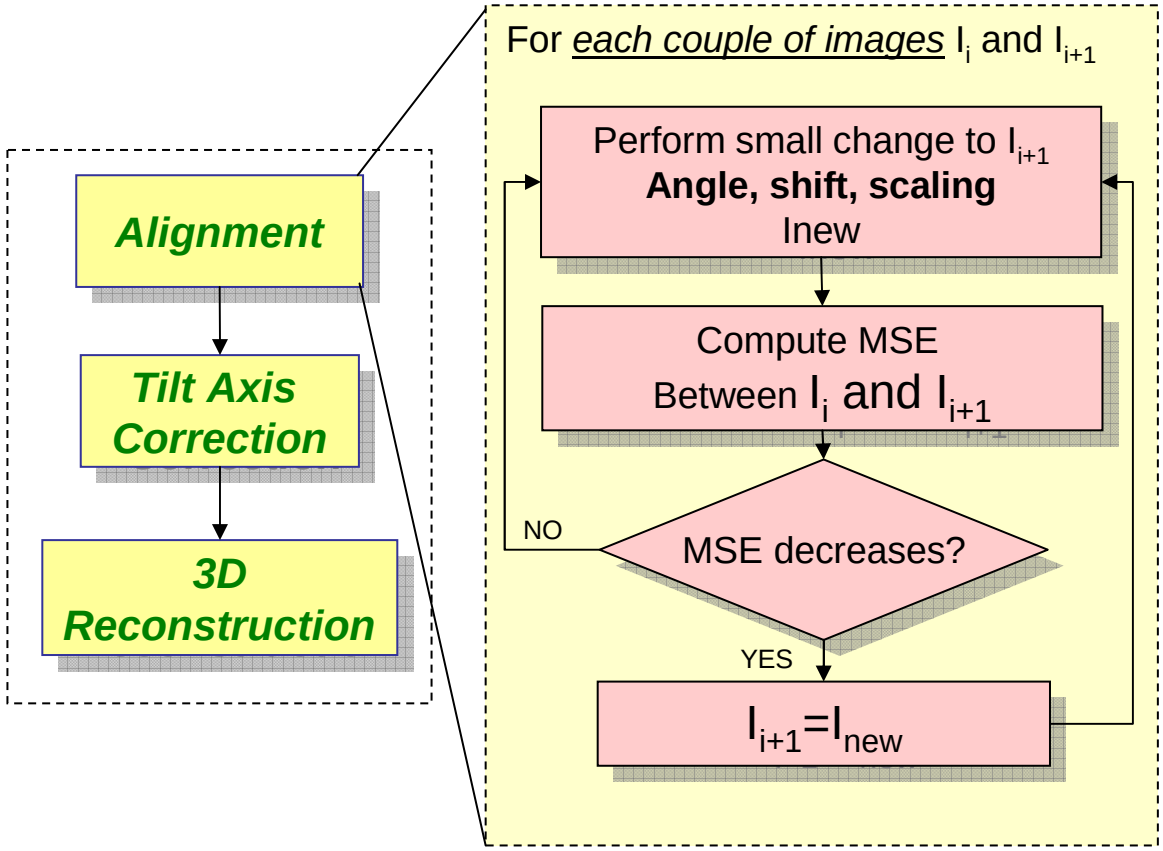
Solution: Alignment of consecutive images



GPU Research

Electron Tomography

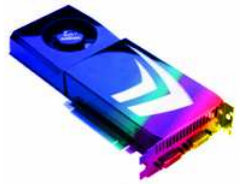




- Local search optimization
- Affine transformations massively used
- 50% of computation time
- Independent optimization for each couple

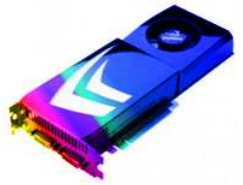
TARGET FUNCTION

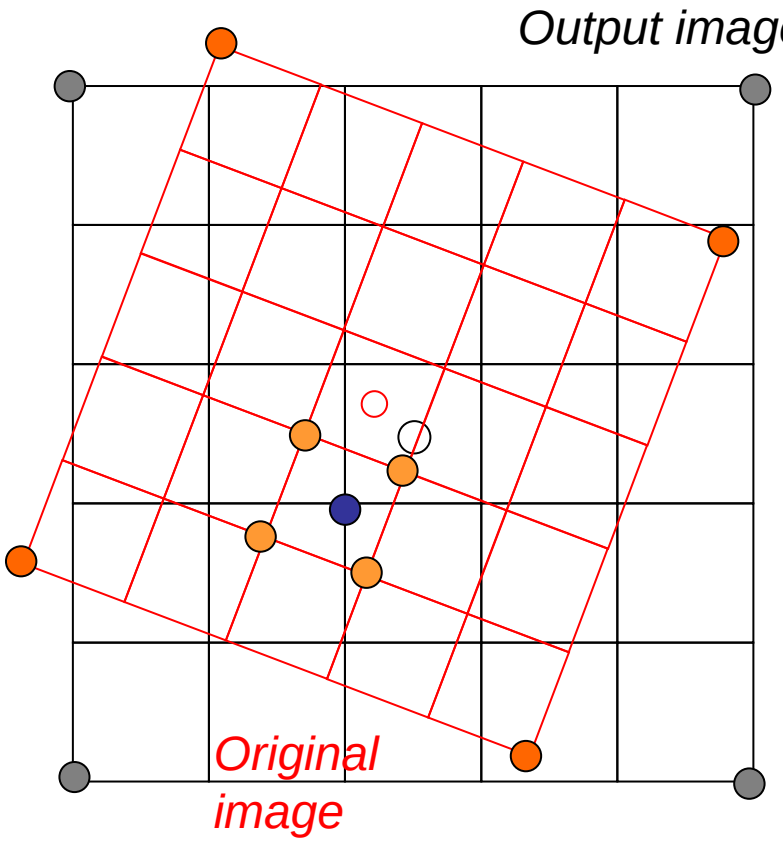
Affine transformation



GPU Research

Affine transformation

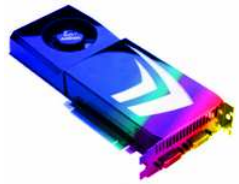


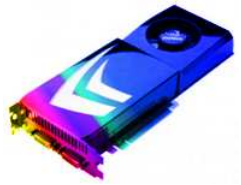


(6x6-pixel images)

- **Rotation, shift and scaling**
- **Pixel intensity is the weighted average of 4 pixels from the original image:**
  - **4 memory reads**
  - **1 write**

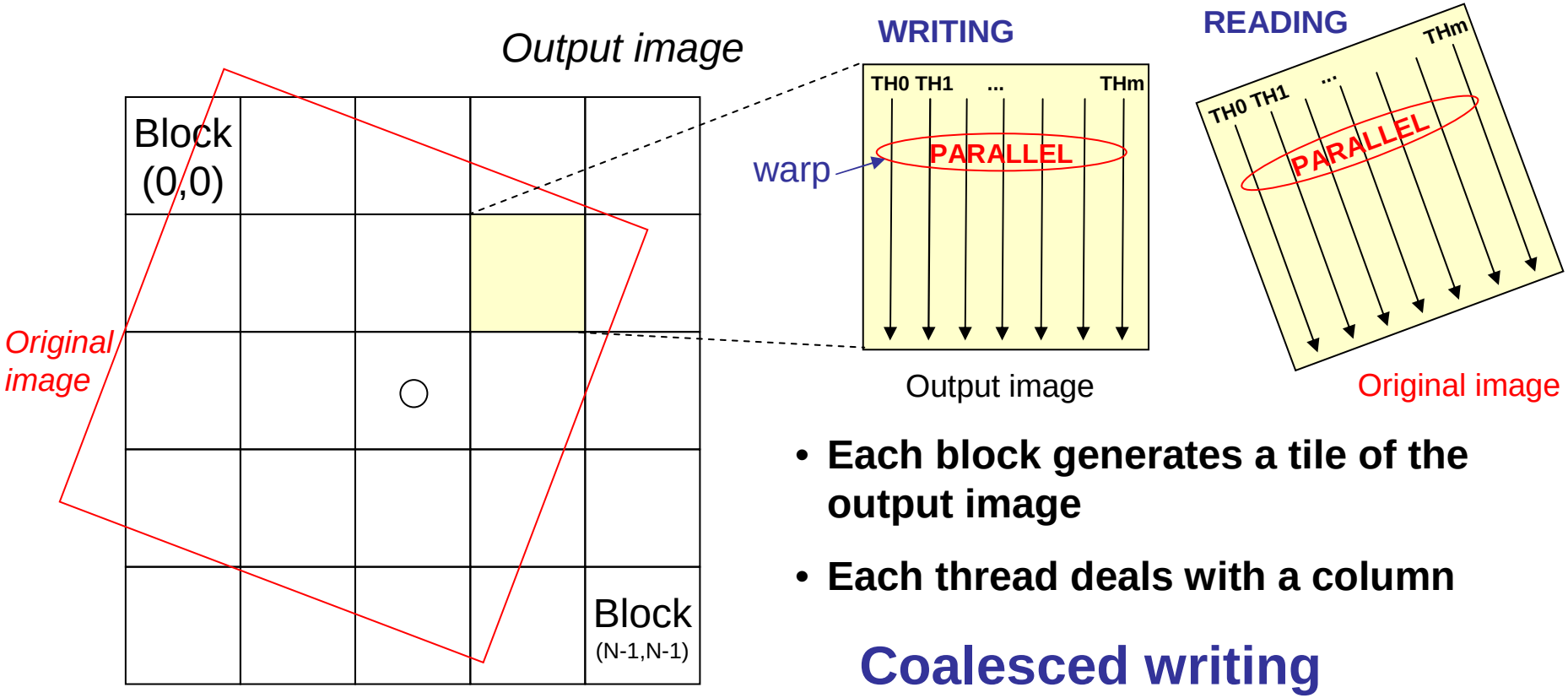
Memory access scheme is the big issue





GPU Research

Affine transformation

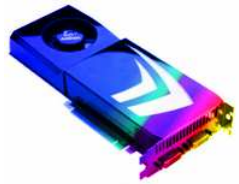


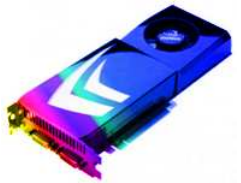
(2048x2048-pixel images)

- **Each block generates a tile of the output image**
- **Each thread deals with a column**

Coalesced writing

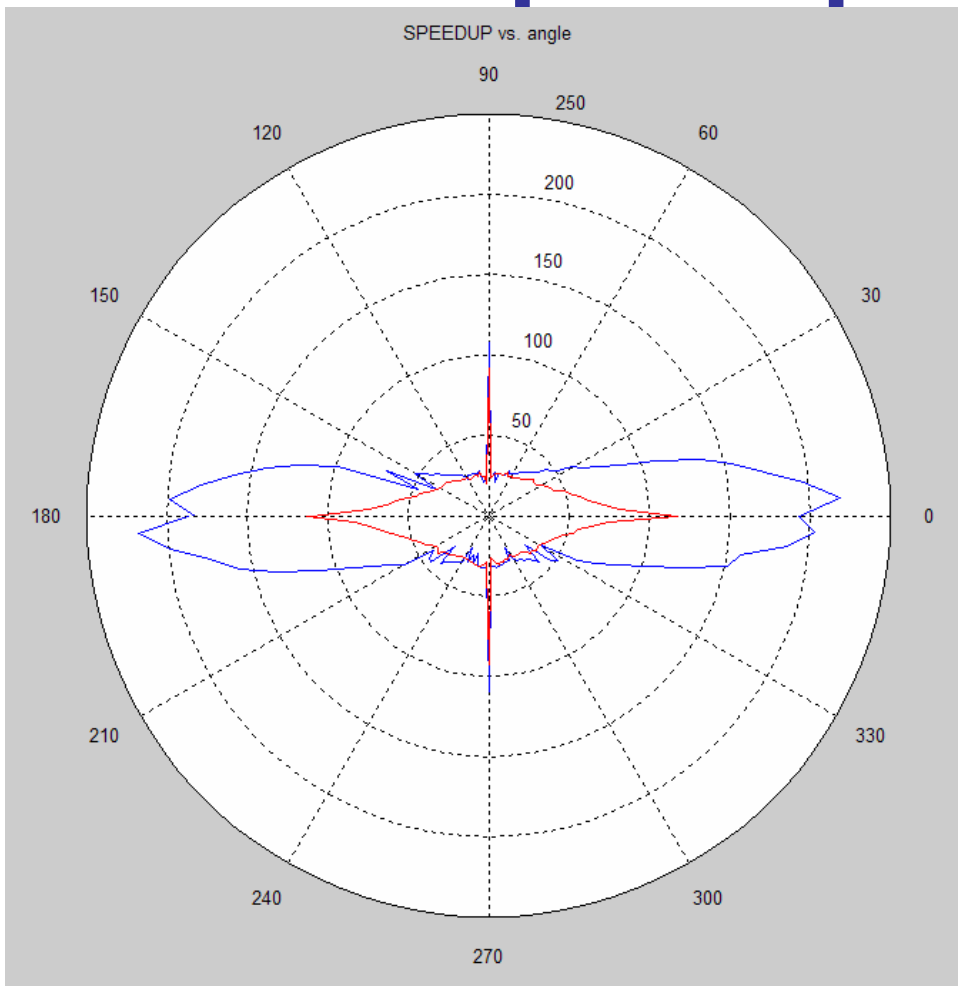
Non-colaesced reading





GPU Research

Speedup vs angle



Baseline 1 core Intel i7

- TESLA C1060 (no cache memory)

  Average x40

FOR OUR APPLICATION:

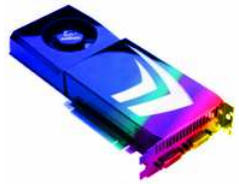
Average ±10º x67

- Geforce GFX480

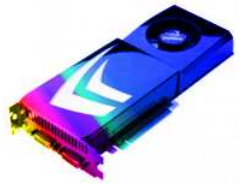
  Average x75

FOR OUR APPLICATION:

Average ±10º= x190



Overall speedup



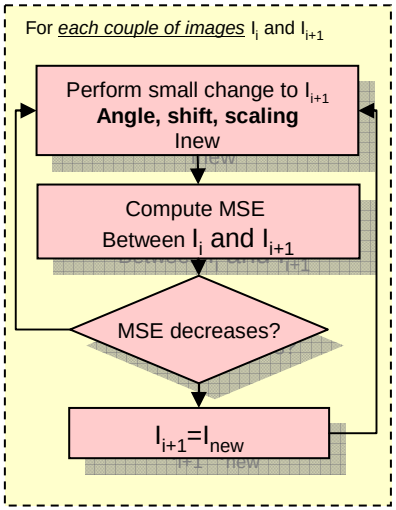
- Average speedup considering the range of angles, shifts and scaling for ET

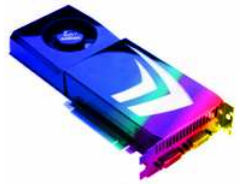
  Rotation [-10º,10º] ; Shift [-300, +300] pixels; Scaling [90%,110%]

DEVICE	Speedup - Only GPU -	Speedup - GPU+Data transfer -
TESLA C1060	×62	×8.5 ⚠
GeForce GTX480	×156	×9.5 ⚠

- Solution:

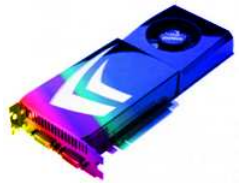
1. Send image I_i and I_{i+1} to GPGPU
2. Start optimization
   1. Call Affine Transformation kernel
   2. Call MSE kernel
   3. Get only MSE from GPGPU
   4. Goto 2.1





GPU Research

What's next



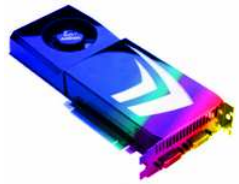
- Include TESLA C2070
- Double-precision Floating-point
  - Unravel *Geforce vs TESLA price difference* mistery

    C2070's ECC memory?
- Use of shared memory?
  - Unnecessary for small angles
- Develop Affine Transformation+MSE kernel
- XMIPP integration
  - Wish us luck!

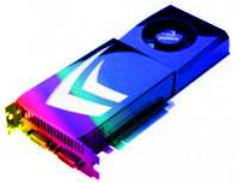
THANKS TO

Eduardo García & Tomás Galán (USP-CEU)

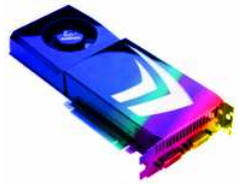
Alessandro Deideri (Polito)



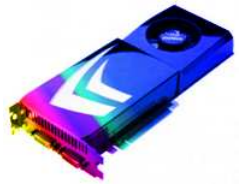
Agenda



- Laboratory of Bioengineering
- HPC, GPUs and FPGAs
- GPGPU research
- FPGA research
- Acknowledgement



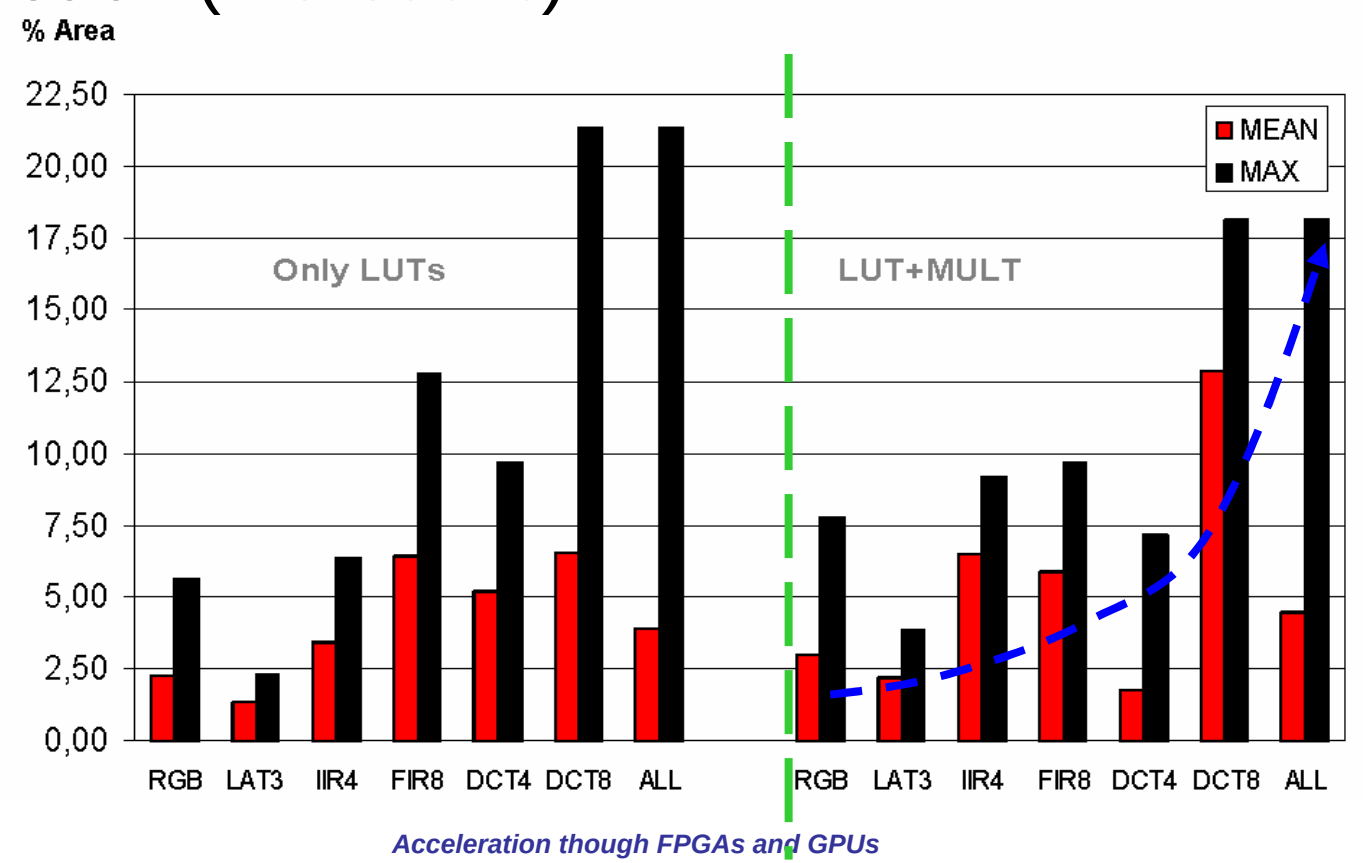
FPGA research



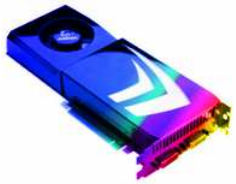
- Implementation of Electron Tomography on FPGA board (Altera DK-DEV-4SGX230N)
- High-Level Synthesis of DSP algorithms
- Automated Precision Optimization
- Custom floating-point and fixed-point/floating-point arithmetic blocks

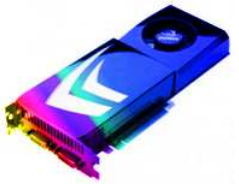
Combined High-Level Synthesis and Fixed-Point Optimization

- Overall area reduction compared to traditional approach (VLSI-SOC'10)



Acceleration though FPGAs and GPUs





FPGA Research

Fixed-Point Error Estimation

PERFORMANCE OF THE ESTIMATION METHOD: PRECISION.

Benchmark	Estimation error [120,100)[1] dB (dB)[2]	(%)[3]	[100,80) dB (dB)	(%)[3]	[80,60) dB (dB)[2]	(%)[3]	[60,40] dB (dB)[2]	(%)[3]
$VEC_{3\times3}$	0.07	0.54	0.07	0.11	0.06	0.50	0.09	0.72
$VEC_{8\times8}$	0.05	0.57	0.04	0.40	0.04	0.57	0.13	1.19
POW^*	0.27	0.98	0.24	0.71	0.29	0.17	0.18	1.52
EQ^*	0.39	5.00	0.17	1.55	0.76	5.96	1.12	12.12
LMS_1^*	0.09	0.41	0.14	0.90	0.16	1.74	0.82	6.96
LMS_5^*	0.09	0.46	0.08	0.07	0.13	1.08	1.09	5.51
VOL_3^*	1.14	3.33	0.49	1.84	0.81	6.70	1.43	16.67
All	0.39	1.27	0.24	0.05	0.76	1.48	1.12	4.21

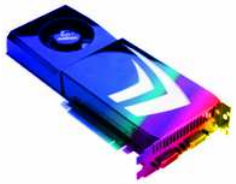
* Recursive

[1] Error constraint

[2] $|10log(\frac{P_{ref}}{P_{est}})|$ (max)

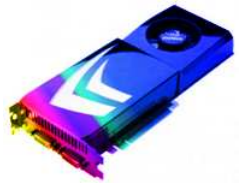
[3] $|100(\frac{P_{ref}-P_{est}}{P_{ref}})|$ (average)

Average error < 5% (VLSI-SOC'10)



FPGA Research

Fixed-Point Error Estimation



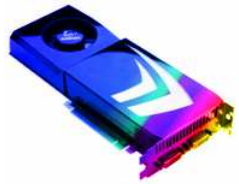
PERFORMANCE OF THE ESTIMATION METHOD: COMPUTATION TIME.

Bench.	FxP Samples	Param. time (secs)+	No. of estimates (mean)	Estimation-based FXP (secs)+	Simulation-based FXP (secs)+	Speed-up
$VEC_{3\times3}$	20000	59.66	150.14	0.03	66.86	×2122
$VEC_{8\times8}$	20000	330.67	1739.96	1.72	2331	×1377
POW^*	20000	546.14	97.15	0.02	21.93	×1048
EQ^*	16000	61.64	231.98	0.12	105.78	×904
LMS_1^*	5000	908.02	712.28	0.42	163.73	×394
LMS_5^*	5000	1646.38	2547.48	7.26	1611.46	×221
VOL_3^*	5000	212.72	673.38	0.29	151.13	×526
All	-	-	-	-	-	×942

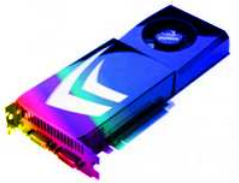
* With feedback

\+ Using 1.66 GHz Intel Core Duo processor and 1 GB of RAM

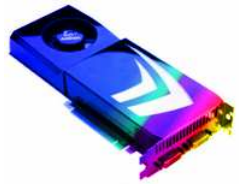
Speedup x942 (VLSI-SOC'10)



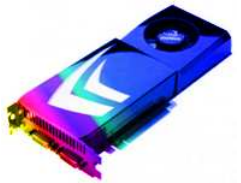
Agenda



- Laboratory of Bioengineering
- HPC, GPUs and FPGAs
- GPGPU research
- FPGA research
- Acknowledgement



Thanks!



- Thanks to my BIOLAB colleagues
  - Carlos Oscar S. Sorzano
  - Abraham Otero Quintana
  - Ana Iriarte Ruíz
- Thanks to our bright students
  - Alessandro Deideri (currently at HP Italy)
  - Tomás Galán García-Obregón
  - Eduardo García de la Cueva

- Our research is currently being funded by
  - Universidad San Pablo-CEU
  - Banco Santander
- Altera has kindly donated
  - PCIe Stratix IV Board
  - Software license
- Nvidia has kindly donated
  - 2x TESLA C1060 boards
  - 1x GeForce GTX 480 board