

# HIGH PERFORMANCE COMPUTING

**BIRD'S EYE VIEW**  
**PROGETTO DI CORSO**

**DI BLASI FABRIZIO**  
**A.A. 2018/2019**

---

# SCOPO PROGETTUALE

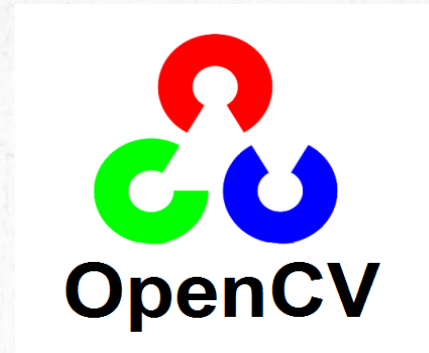
- Trasformazione omografica dei punti di un'immagine in un piano orientato a piacere. Tale tipologia di proiezione è molto utilizzata nel campo della visione artificiale, in particolare in ambito automotive per la misura della distanza dagli ostacoli
- Ricerca delle zone soggette a più carico computazionale e fornire una parallelizzazione
- Benchmarking dell'algoritmo e calcolo dello speedup in dispositivi diversi general purpose
- Benchmarking su Nvidia Jetson Nano





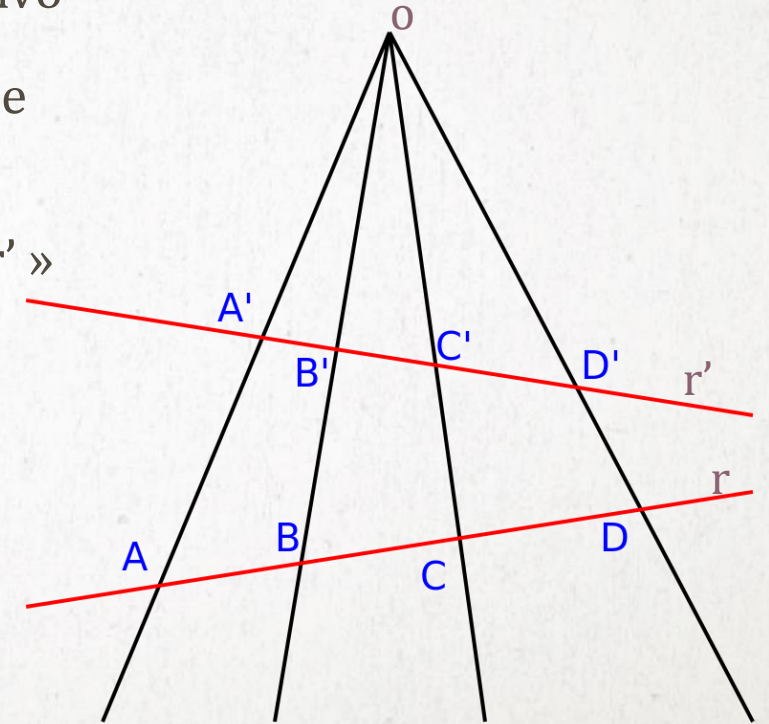
# LIBRERIE UTILIZZATE

- OpenCV
- CUDA
- FFmpeg

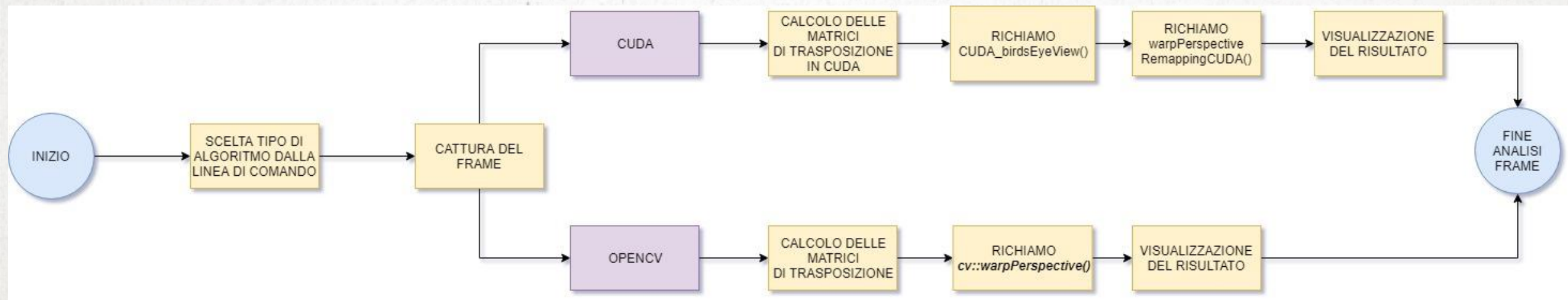


# CENNI MATEMATICI

- L'omografia è una particolare trasformazione lineare in cui a ciascun punto dello spazio di partenza corrisponde un solo punto in quello di arrivo
- Ai fini di scopi progettuali, si suppone il punto di osservazione in posizione fissate
- L'utente ha la possibilità di variare l'inclinazione del piano «  $r'$  » a suo piacimento, e la distanza tra il punto di osservazione «  $o$  » ed il piano «  $r$  »



# SCHEMA A BLOCCHI





# INTERFACCIA UTENTE

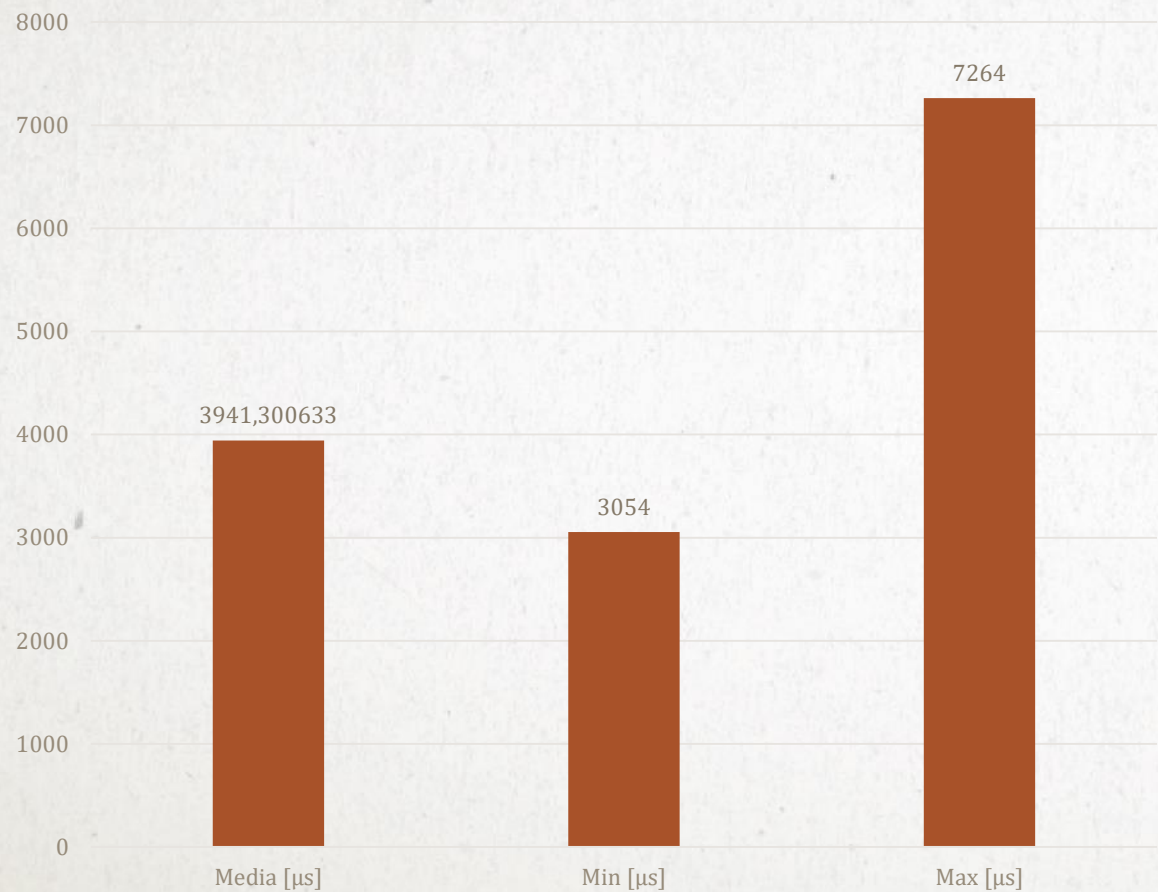
- L'utente ha a disposizione 5 diverse regolazioni
- Le prime tre impostano l'inclinazione del nuovo piano
- $f$  : apertura focale della fotocamera
- Distance : distanza del punto di osservazione
- Esecuzione con soli kernel CUDA  
\$ ./app y  
\$ ./app y <video path>
- Esecuzione con solo chiamate ad OpenCV:  
\$ ./app n  
\$ ./app n <video path>



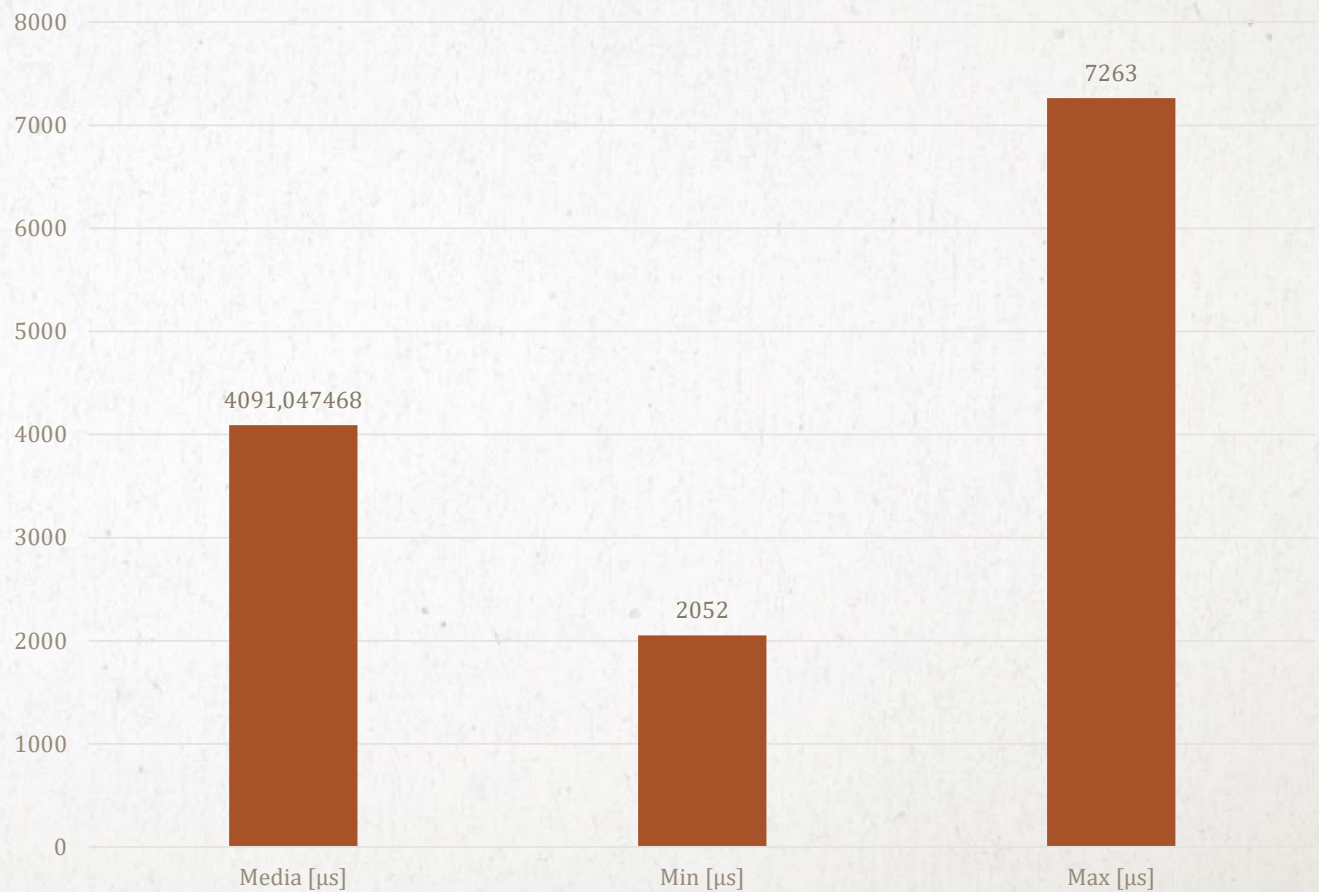
# CONFRONTO DEI BENCHMARK (1)

WORKSTATION : AMD® RYZEN 7 2700X EIGHT-CORE PROCESSOR × 16 THREAD @ 4.3GHZ  
GEFORCE GTX 750 2GB TI/PCIE/SSE2  
16 GB RAM 3200MHZ DUAL CHANNEL

Benchmark Workstation - CUDA



Benchmark Workstation - OpenCV





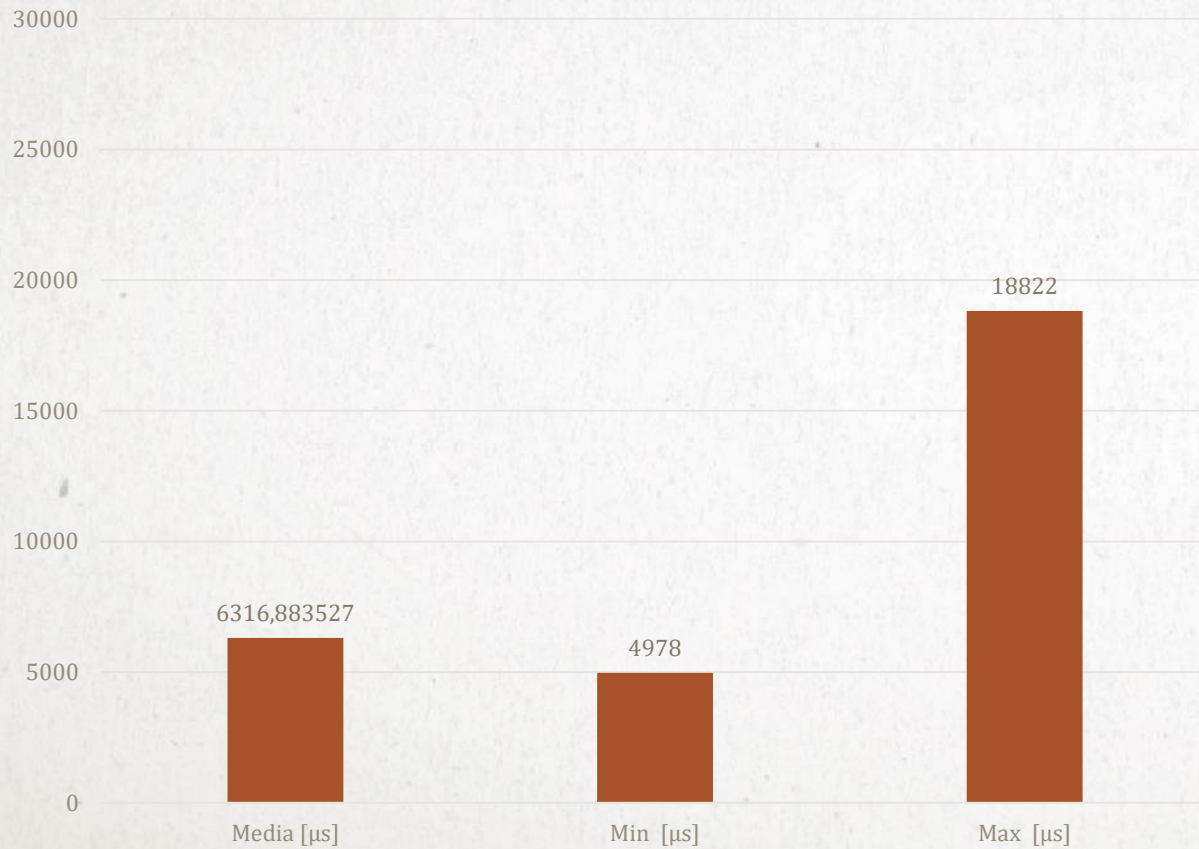
# CONFRONTO DEI BENCHMARK (2)

LAPTOP : INTEL® CORE™ I5-6200U CPU @ 2.30GHZ × 4

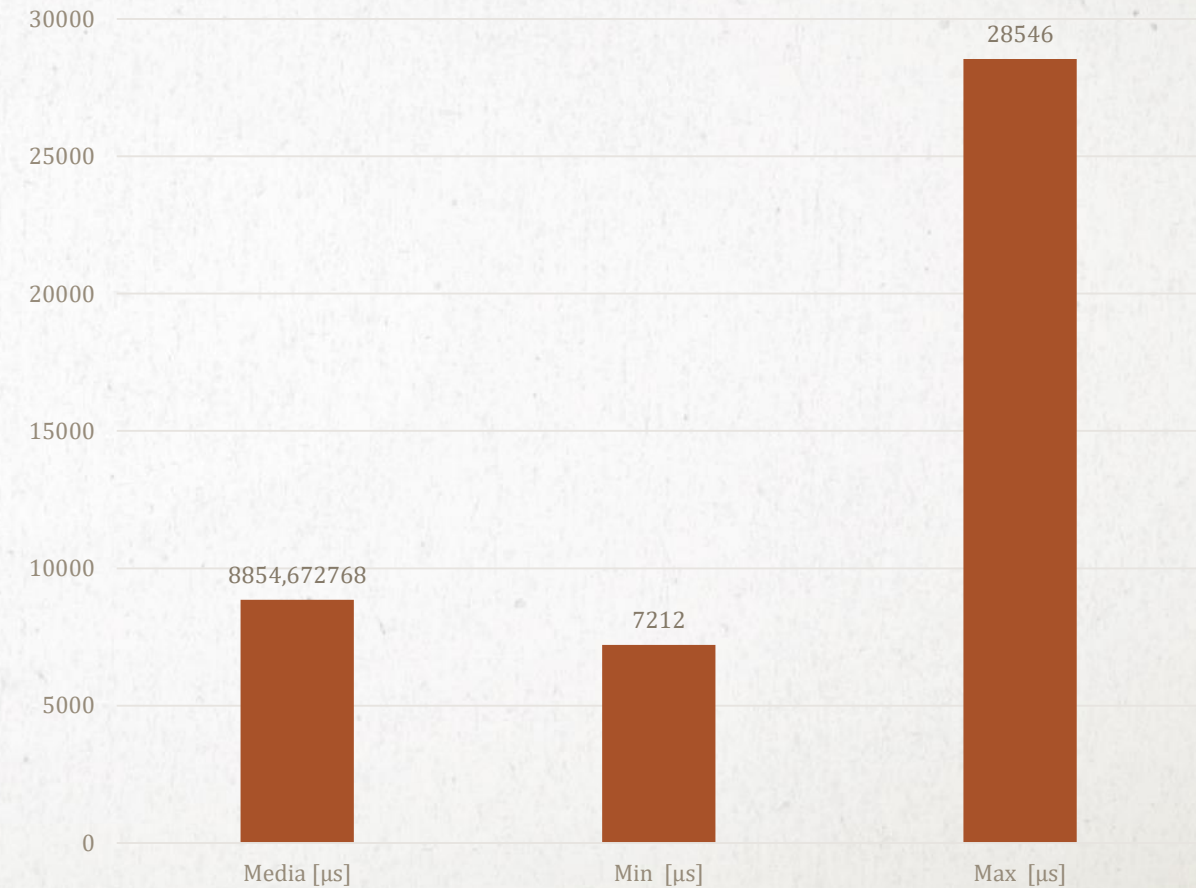
GEFORCE 920M 2GB /PCIE/SSE2

8 GB RAM 1600MHZ DUAL CHANNELL

Benchmark Laptop - CUDA



Benchmark Laptop - OpenCV





# CONFRONTO DEI BENCHMARK (2)

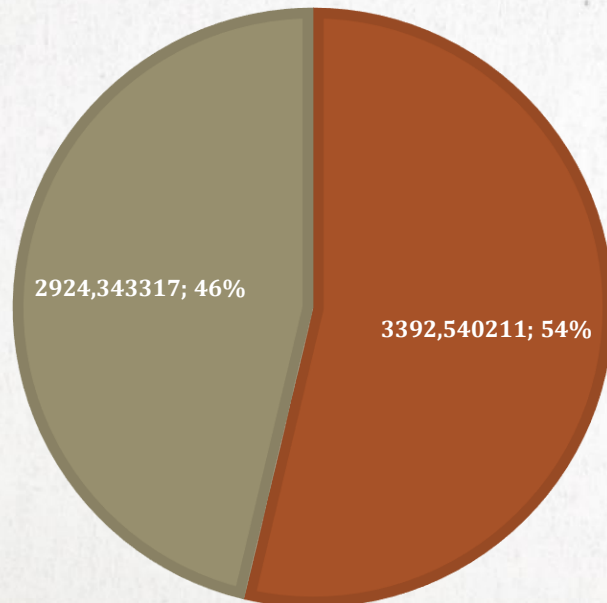
**LAPTOP :** INTEL® CORE™ I5-6200U CPU @ 2.30GHZ × 4

GEFORCE 920M 2GB /PCIE/SSE2

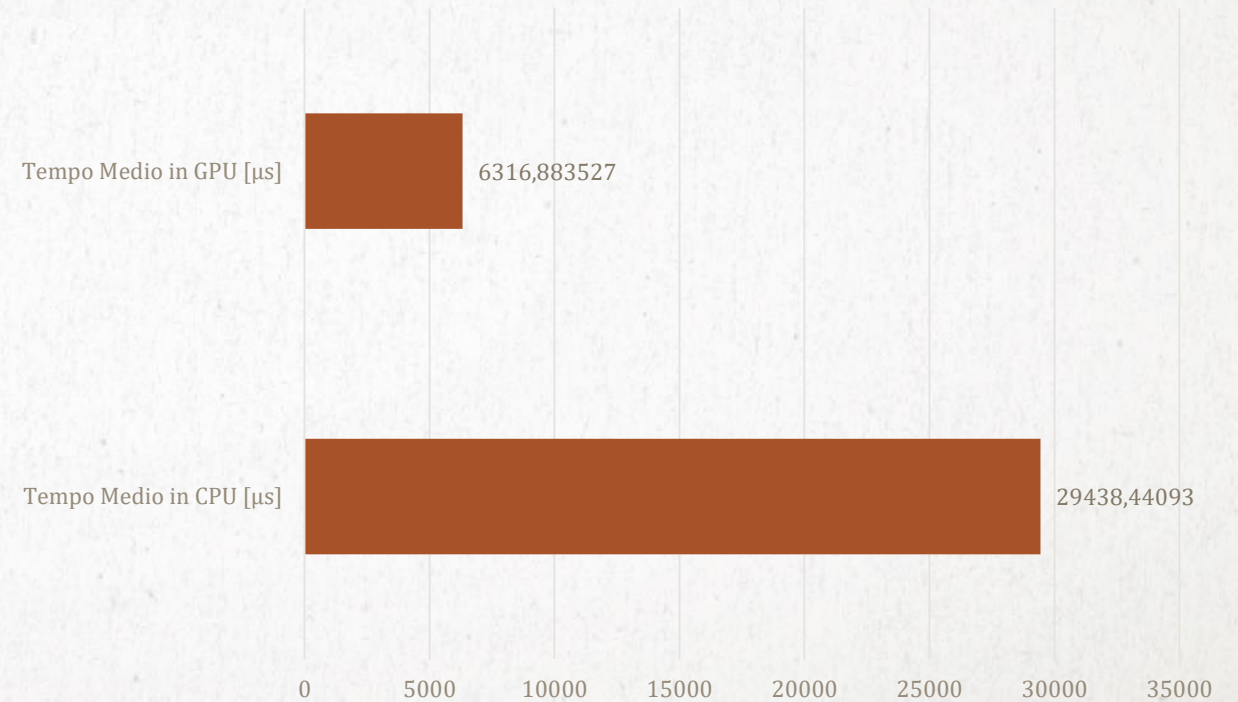
8 GB RAM 1600MHZ DUAL CHANNELL

## COMPOSIZIONE TEMPO MEDIO ANALISI FRAME CUDA

■ tempo warping medio [μs]   ■ Restante parte computazione [μs]



## Confronto tempi CPU / GPU



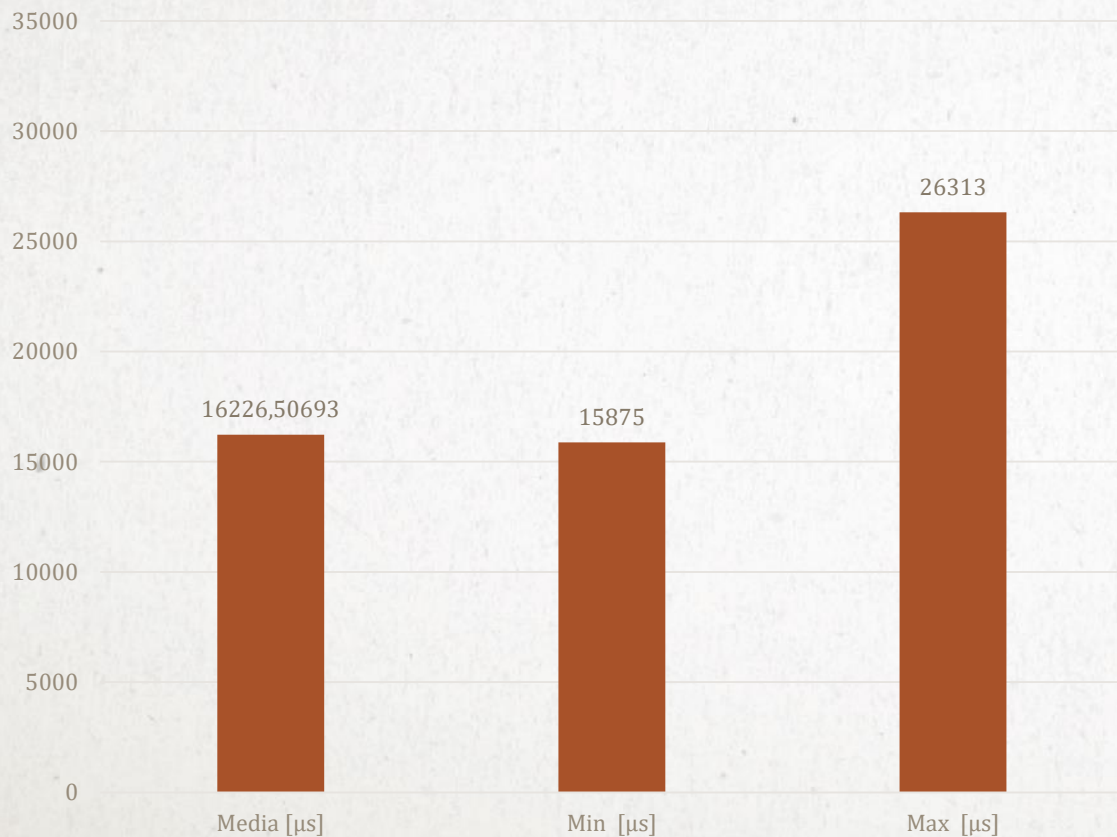
# CONFRONTO DEI BENCHMARK (3)

**JETSON NANO : 64-BIT QUAD-CORE ARM A57 @ 1.43GHZ**

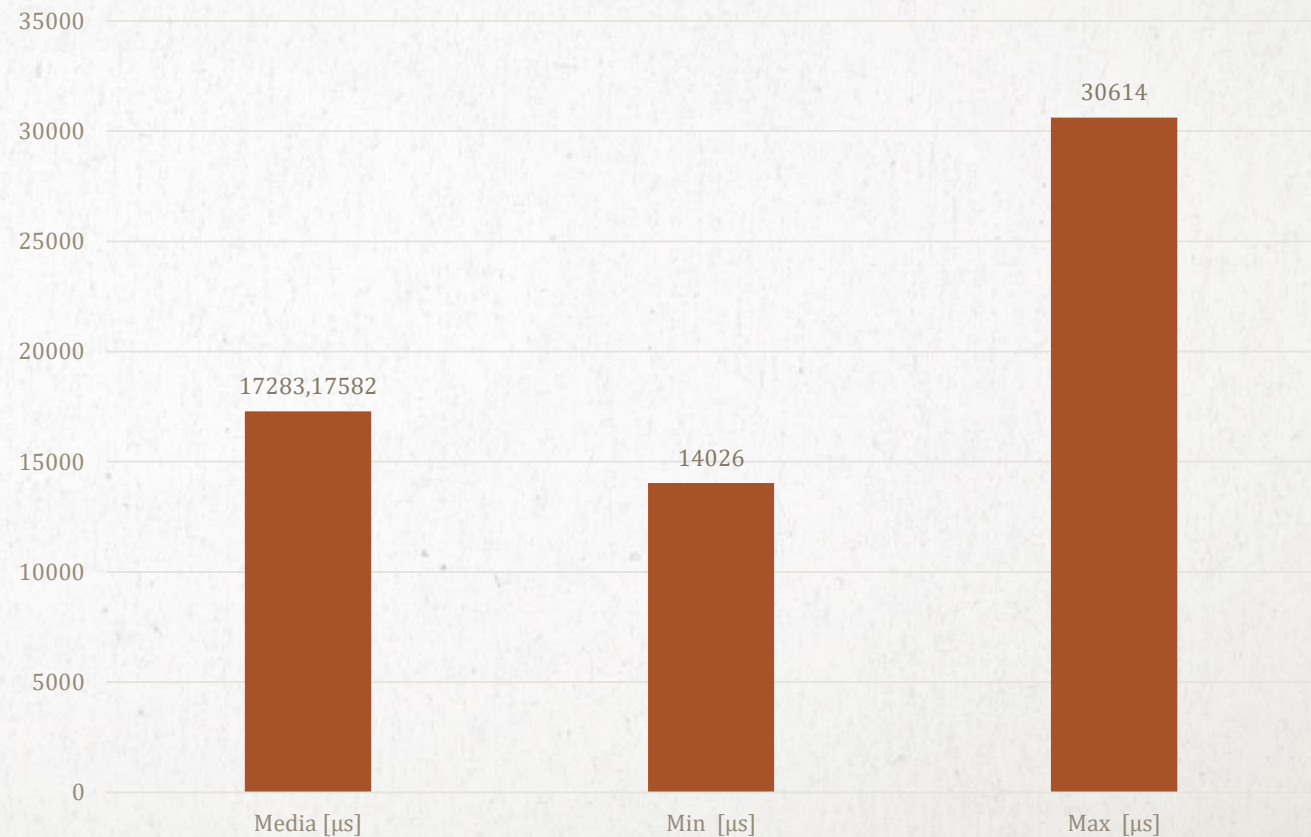
**128-CORE NVIDIA MAXWELL @ 921MHZ**

**4GB 64-BIT LPDDR4 @ 1600MHZ | 25.6 GB/S**

Jetson – Tempi Medi Per Singolo Frame - OpenCV



Jetson - Tempi Medi Per Singolo Frame - CUDA





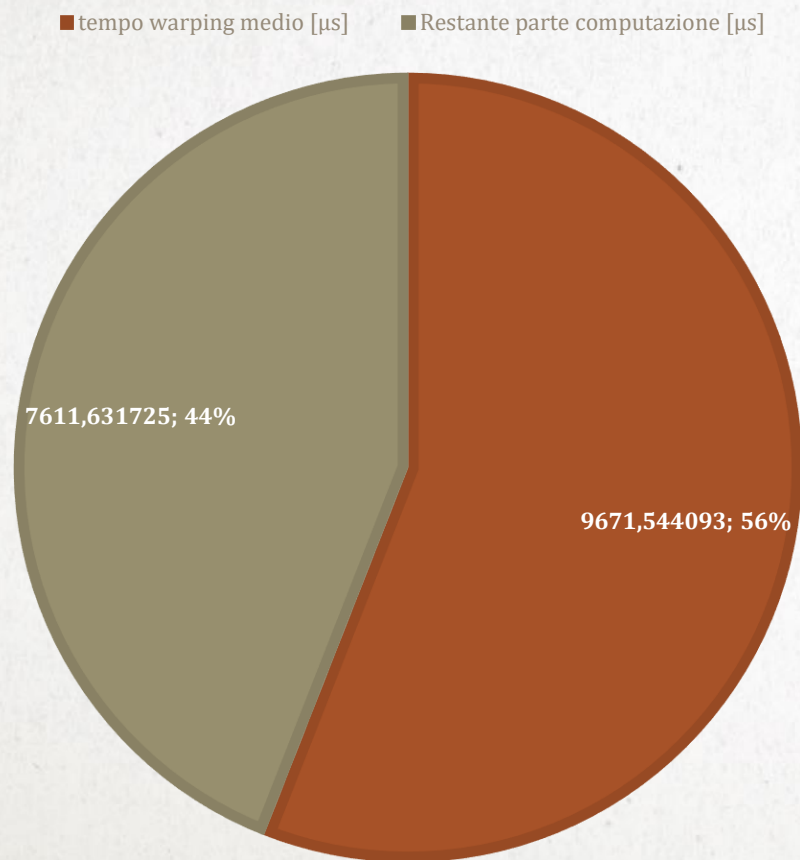
# CONFRONTO DEI BENCHMARK (3)

JETSON NANO : 64-BIT QUAD-CORE ARM A57 @ 1.43GHZ

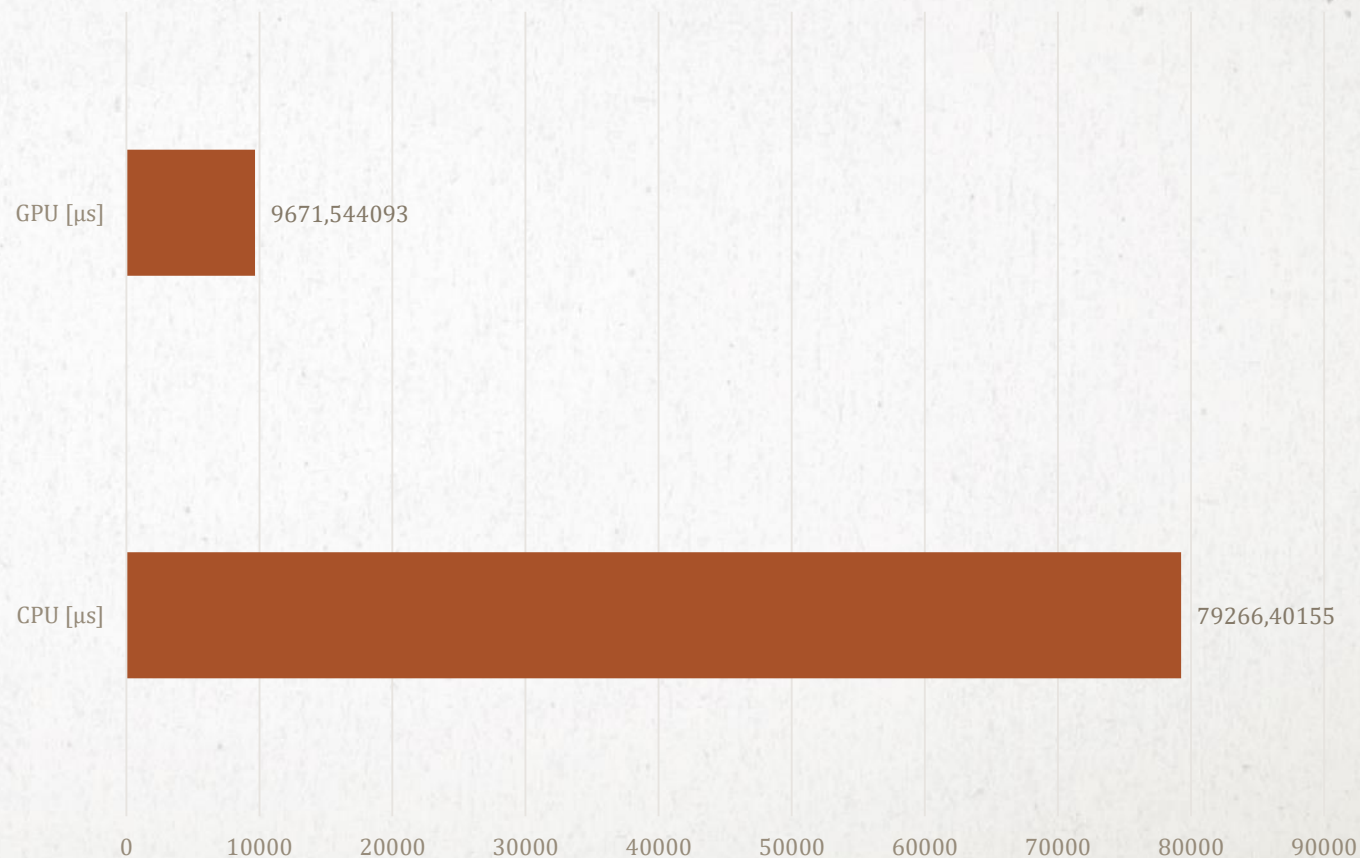
128-CORE NVIDIA MAXWELL @ 921MHZ

4GB 64-BIT LPDDR4 @ 1600MHZ | 25.6 GB/S

## COMPOSIZIONE TEMPO MEDIO ANALISI FRAME CUDA



## CONFRONTO TRA CPU / GPU



# CONCLUSIONI SUI BENCHMARK (1)

- Definendo lo *SpeedUp* come  $\frac{\text{Media tempi algoritmo OpenCV}}{\text{Media tempi algoritmo CUDA}}$  si ottiene che:
- Nella workstation risulta un rapporto di 0.98, ciò significa che le due versioni quasi si equivalgono, ma quella utilizzando solo OpenCV è leggermente più veloce
- Nel laptop, invece, si ottiene un rapporto di 1,40, il che è un ottimo risultato, ma calcolando invece il rapporto tra  $\frac{\text{Media tempi algoritmo CPU}}{\text{Media tempi algoritmo CUDA}}$  si ottiene 4,66
- Per il Jetson, invece, calcolando  $\frac{\text{Media tempi algoritmo OpenCV}}{\text{Media tempi algoritmo CUDA}}$  si ottiene 0,9388  
invece calcolando  $\frac{\text{Media tempi algoritmo CPU}}{\text{Media tempi algoritmo CUDA}}$  si ha 8,1692



## CONCLUSIONI SUI BENCHMARK (2)

- La procedura di offload del calcolo matriciale è controproducente
  - Effettuando le moltiplicazioni in CPU aumenta molto di più lo SpeedUP tra versione CPU e GPU «ibrida» dello stesso algoritmo:
  - Laptop : 7,65
  - Jetson : 11,19
-

**GRAZIE PER  
L'ATTENZIONE**