UNIONE EUROPEA

REGIONE PUGLIA
AREA POLITICHE PER LO SVILUPPO IL LAVORO E L'INNOVAZIONE

AIUTI A SOSTEGNO DEI CLUSTER TECNOLOGICI REGIONALI PER L'INNOVAZIONE

PERSON

**PERSON** - PERVASIVE GAME FOR PERSONALIZED TREATMENT OF COGNITIVE AND FUNCTIONAL DEFICITS ASSOCIATED WITH CHRONIC AND NEURODEGENERATIVE DESEASES

# Deliverable D10.3 – Progettazione del set-up di calcolo GP-GPU based, degli algoritmi e della codifica per GP-GPU

| | | |
|---|---|---|
| **Codice Progetto** | : | LQ8FBY0 |
| **Titolo Progetto** | : | PERSON |

| | | |
|---|---|---|
| **Numero Deliverable** | : | D10.3 |
| **Titolo Deliverable** | : | Progettazione del set-up di calcolo GP-GPU based, degli algoritmi e della codifica per GP-GPU |
| **Natura del Deliverable** | : | Report |
| **Mese di Rilascio da Capitolato** | : | Marzo 2017 |
| **Livello di Disseminazione** | : | Pubblico |
| **Versione** | : | 1.0 |
| **Data Deliverable** | : | 31/03/2017 |


# Abstract

Dopo aver presentato le caratteristiche delle unità di calcolo utilizzate, viene descritto il flusso di acquisizione e di elaborazione. Si esaminano quindi le prestazioni ottenute utilizzando le diverse unità di calcolo a disposizione.

Infine, si descrivono le strategie di ottimizzazione degli algoritmi applicate mettendo a confronto il loro impatto sulle prestazioni.



# Keyword list

GPU; CUDA; elaborazione di immagini; calcolo parallelo; prestazioni di elaborazione;


# Tracking delle Versioni

| **Version** | **Changes** | **Author(s)** |
|---|---|---|
| **1.0** | **First Draft** | **Maria Francesca de Ruvo, Marco Colaprico** |

# Indice

**1. POTENZIALITÀ DEL GPU COMPUTING ..... 6**

**2. CARATTERISTICHE DELLE UNITÀ DI CALCOLO UTILIZZATE ..... 6**

**3. GESTIONE ED OTTIMIZZAZIONE DEI FLUSSI DI ACQUISIZIONE E DI ELABORAZIONE ..... 7**

**4. GESTIONE DELLA MEMORIA PER LA CONDIVISIONE DATI PC-GPU ..... 12**

4.1 UTILIZZO STANDARD DELLA COMUNICAZIONE MEMORIA PC-MEMORIA GPU ..... 13

4.2 DEFINIZIONE STREAM DI ELABORAZIONE SU GPU ..... 13

4.3 UTILIZZO DEL MAPPING TRA MEMORIA DEL PC E SPAZIO DEGLI INDIRIZZI DELLA GPU ..... 14

4.4 IMPATTO SULLE PERFORMANCE E CONFRONTO DEI RISULTATI ..... 14

**5. OTTIMIZZAZIONE DELL'ELABORAZIONE GPU ..... 15**

5.1 ALGORITMI ..... 15

5.2 GESTIONI DELLA MEMORIA ..... 16

**APPENDICE 1 – STRUTTURA DEL LOG DI ELABORAZIONE CON CPU ..... 18**

**APPENDICE 2 – STRUTTURA DEL LOG DI ELABORAZIONE CON GPU NVIDIA TESLA K40 ..... 18**

**APPENDICE 3 – STRUTTURA DEL LOG DI ELABORAZIONE CON GPU NVIDIA GEFORCE TITAN ..... 19**

**APPENDICE 4 – STRUTTURA DEL LOG DI ELABORAZIONE CON GPU NVIDIA TESLA K40 E MEMORIA PAGINABILE ..... 21**

**APPENDICE 5 – STRUTTURA DEL LOG DI ELABORAZIONE CON GPU NVIDIA TESLA K40 E "ZERO COPY MEMORY" ..... 22**

**APPENDICE 6 – VIDEO ELABORATI ..... 23**

**APPENDICE 7 – CODICE SORGENTE ..... 23**

**BIBLIOGRAFIA ..... 24**

# Indice figure

Figura 1. Componenti del sistema ..... 7
Figura 2. Diagramma di sequenza ..... 8
Figura 3. Flusso di elaborazione ..... 9
Figura 4. Confronto dei tempi di elaborazione delle coppie di immagini ..... 13
Figura 5. Utilizzo della GPU con memoria paginabile ..... 16
Figura 6. Utilizzo della GPU con memoria "page-locked" ..... 16
Figura 7. Utilizzo della GPU con memoria "zero copy" ..... 17

# Indice grafici

Grafico 1. Tempi di elaborazione con GPU Tesla K40 ..... 9
Grafico 2. Tempi di elaborazione con GPU Geforce GTX Titan ..... 10
Grafico 3. Tempi di elaborazione con CPU ..... 10
Grafico 4. Percentuale delle immagini elaborate con GPU Tesla K40 ..... 11
Grafico 5. Percentuale delle immagini elaborate con GPU Geforce GTX Titan ..... 11
Grafico 6. Percentuale delle immagini elaborate con CPU ..... 12
Grafico 7. Tempi di elaborazione con GPU e uso della memoria paginabile ..... 14
Grafico 8. Tempi di elaborazione con GPU e uso della memoria "page-locked" ..... 15
Grafico 9. Tempi di elaborazione con GPU e uso della memoria "zero copy" ..... 15

## Glossario, acronimi e abbreviazioni

| Item | Descrizione |
|---|---|
| SDVA | Set up di visione artificiale |
| 3D | Tridimensionale |
| GB | Giga bytes |
| GB/s | Giga bytes al secondo |
| CPU | Central Processing Unit |
| GPU | Graphical Processing Unit |
| GP-GPU | General Purpose GPU |
| CUDA | Compute Unified Device Architecture, architettura hardware di tipo SIMD per l'elaborazione parallela creata da NVIDIA |
| SIMD | Single Instruction Multiple Data, paradigma di architettura di calcolo parallela che prevede che lo stesso flusso di istruzioni sia operato su flussi multipli di dati |
| PCIe | Peripheral Component Interconnect Express, standard di interfaccia d'espansione a bus seriale per computer usato per connettere periferiche alla scheda madre |
| GDDR5 | GDDR, acronimo per Graphics Double Data Rate, è una tecnologia di memoria RAM specifica per schede video. Una delle caratteristiche di questa tipologia di memoria è l'utilizzo di due segnali di clock (uno per l'invio dei comandi e l'altro per le operazioni di lettura e scrittura) permette di inviare dati a più di 5 Gbps minimizzando il rumore |
| FLOPS | FLoating point Operations Per Second e indica il numero di operazioni in virgola mobile eseguite in un secondo dall'unità do calcolo |
| ms | Millisecondi (1ms = $10^{-3}$ secondi) |

# 1. Potenzialità del GPU computing

Il GPU Computing usufruisce dell'unità di elaborazione grafica come co-processore per accelerare operazioni di calcolo "general purpose". L'architettura computazionale SIMD (Single Instruction Multiple Data) permette di eseguire in maniera efficiente non solo istruzioni di computer grafica (i.e. le tipiche operazioni che una scheda video elabora in un PC) ma anche operazioni matematiche su grandi quantità di dati.

Dal 2007 si è sviluppato un considerevole trend tecnologico e commerciale che ha portato all'evoluzione delle unità di elaborazione grafica (GPU); oggi costituiscono i nodi di calcolo dei più potenti computer al mondo (e.g. Titan Supercomputer presso Oak Ridge National Lab, Piz Daint presso the Swiss National Supercomputing Centre) [1].

Diverse tecniche di elaborazione delle immagini prevedono implementazioni efficienti per GPU. Non sempre l'utilizzo di questa tipologia di architettura computazionale introduce un miglioramento delle prestazioni, al contrario se non programmate correttamente possono produrre prestazioni peggiori.

# 2. Caratteristiche delle unità di calcolo utilizzate

La CPU utilizzata per il confronto con le schede video è un processore Intel Xeon E5-2650 v3 con le seguenti caratteristiche:

- 10 core
- 20 thread
- Frequenza massima 3 GHz
- 25 MB di memoria cache
- Larghezza di banda di memoria massima par a 68 GB/s

Come GPU sono stati utilizzati due modelli NVIDIA: Geforce GTX Titan e Tesla K40. La Geforce GTX Titan equipaggiata con processore GK110 con architettura Kepler (capability 3.5) (Figura 10.A), che presenta le seguenti caratteristiche:

- 2688 CUDA core
- Clock 876 Mhz
- 6 GB di memoria GDDR5 con banda passante di 288 GB/s
- Bus PCIe 3.0
- Perfomance di picco in virgola mobile sono di 4.5 Tflops (singola precisione) e 1.5 Tflops (doppia precisione)

La Tesla K40 equipaggiata con processore GK110 con architettura Kepler (capability 3.5). Secondo datasheet dispone di:

- 2880 CUDA core
- Clock 875 Mhz
- 12 GB di memoria GDDR5 con banda passante pari a 288 GB/s
- Bus PCIe 3.0
- Perfomance di picco in virgola mobile sono di 4.29 Tflops (singola precisione) e 1.43 Tflops (doppia precisione)

L’architettura Kepler offre streaming multiprocessor più potenti rispetto all’architettura precedente (Fermi) e la possibilità di eseguire simultaneamente fino a 32 connessioni (gestite in hardware) tra CPU e GPU. Gli stream ad esempio possono essere eseguite contemporaneamente e non più semplicemente sovrapposti.

## 3. Gestione ed ottimizzazione dei flussi di acquisizione e di elaborazione

Nel caso in esame, il flusso di elaborazione segue il seguente schema:

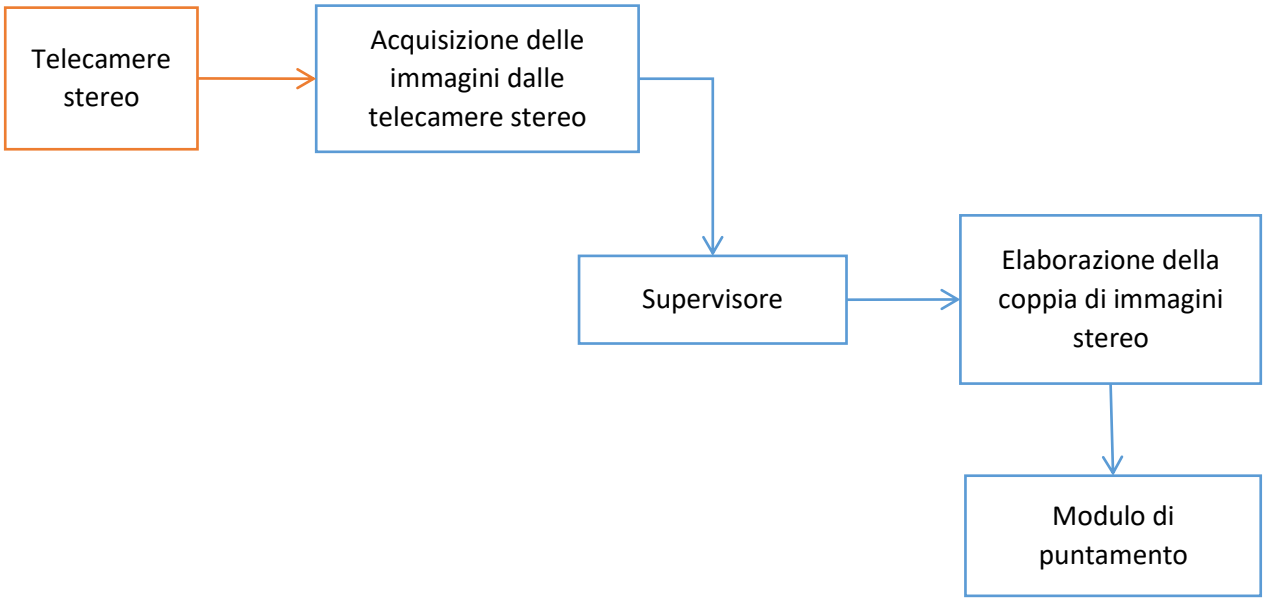


**Figura 1. Componenti del sistema**

Nelle elaborazioni in tempo reale la durata di un ciclo di elaborazione deve essere minore o uguale al periodo di acquisizione; quest’ultimo è costante ed imposto dalla sorgente dell’informazione, mentre la durata di un ciclo di elaborazione dipende dal contenuto informativo acquisito istante per istante.

Il processo di acquisizione è demandato ad una risorsa esterna (la coppia di telecamere stereo) la cui frequenza di acquisizione è impostata a 50 Hz. Il modulo di puntamento deve fornire le coordinate dell’area del monitor indicata alla stessa frequenza, per cui è necessario che l’elaborazione richieda meno di 20 ms.

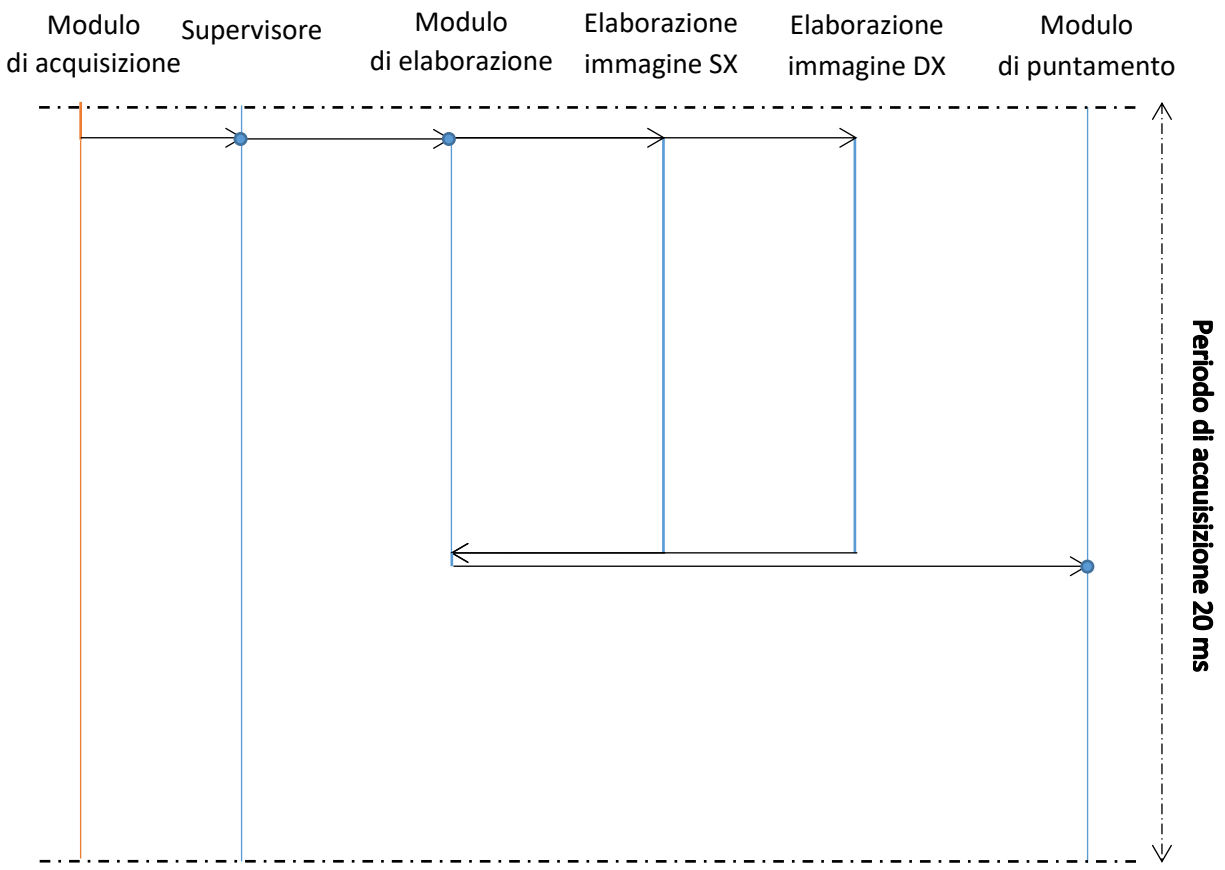


**Figura 2. Diagramma di sequenza**

I tre componenti (i.e. modulo di acquisizione, supervisore e modulo di puntamento) sono sempre in esecuzione, a differenza del modulo di elaborazione che viene eseguito non appena le immagini sono acquisite e trasferite in memoria. A seguito dell'elaborazione delle due immagini è necessario imporre un punto di sincronizzazione poiché servono entrambi i risultati per ottenere la ricostruzione 3D dei marker individuati (nonché il pixel indicato sul monitor).

L'area di memoria in cui vengono copiate le immagini acquisite è condivisa tra il supervisore ed i moduli di elaborazione onde evitare copie ridondanti. Per prevenire conflitti di scrittura/lettura qualora l'elaborazione dovesse richiedere più di un periodo di acquisizione, il supervisore evita di elaborare la coppia di immagini acquisite se è ancora in corso l'elaborazione della coppia precedente. In questo modo si evita anche la saturazione del buffer di acquisizione e non essendoci una coda delle coppie di immagini perse, non appena si conclude l'elaborazione, il supervisore fornirà al modulo di calcolo l'ultima coppia di immagini acquisita.

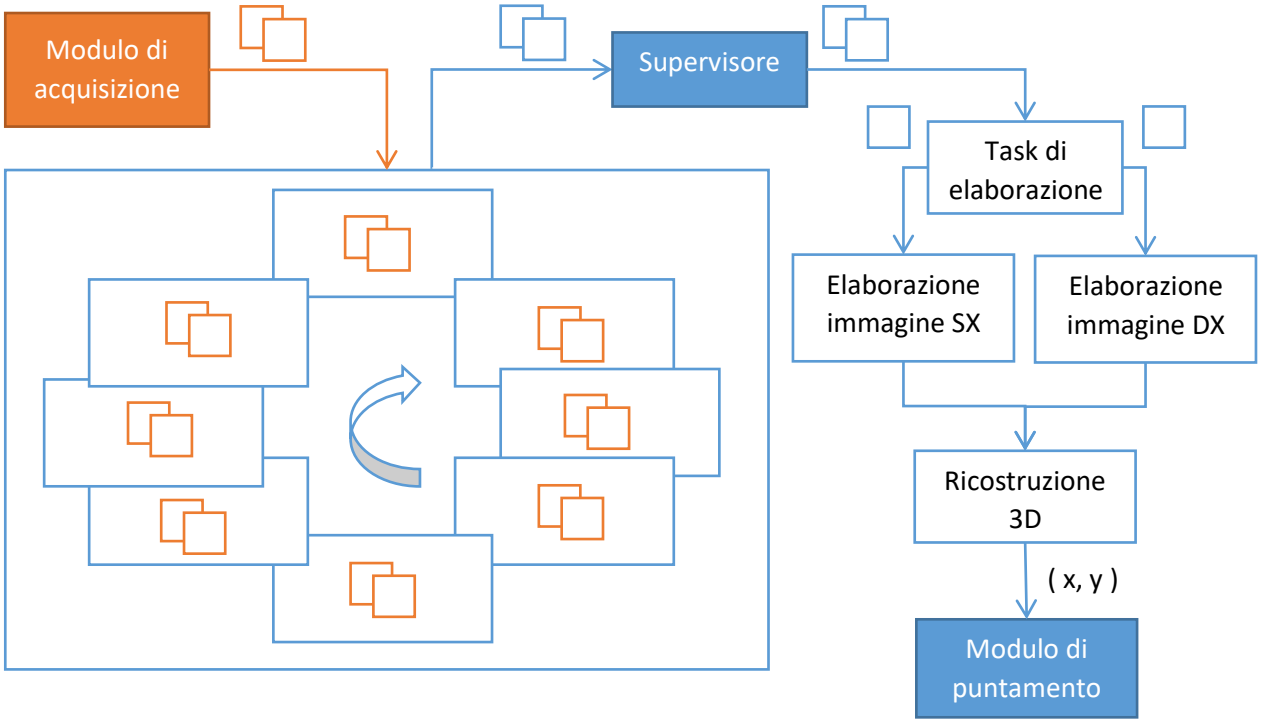


**Figura 3. Flusso di elaborazione**

Dal punto di vista del modulo di puntamento, in caso di sovraccarico delle code dei processi di elaborazione, non si avrà un "effetto ritardo" del puntamento ma bensì un repentino inseguimento della traiettoria descritta dall'utilizzatore.

L'utilizzo della GPU ha portato ad un notevole miglioramento delle performance raggiungendo un tempo di elaborazione pari a 10,99 ms e deviazione standard pari a 0,89 ms (per la Tesla K40, 13,28 ms con deviazione standard pari a 1,50 ms per la GTX Titan), a differenza di quanto ottenuto con il solo utilizzo della CPU che richiede un tempo di elaborazione pari a 19,01 ms e deviazione standard pari a 10,05 ms, non in grado di soddisfare la richiesta di elaborazione in tempo reale.

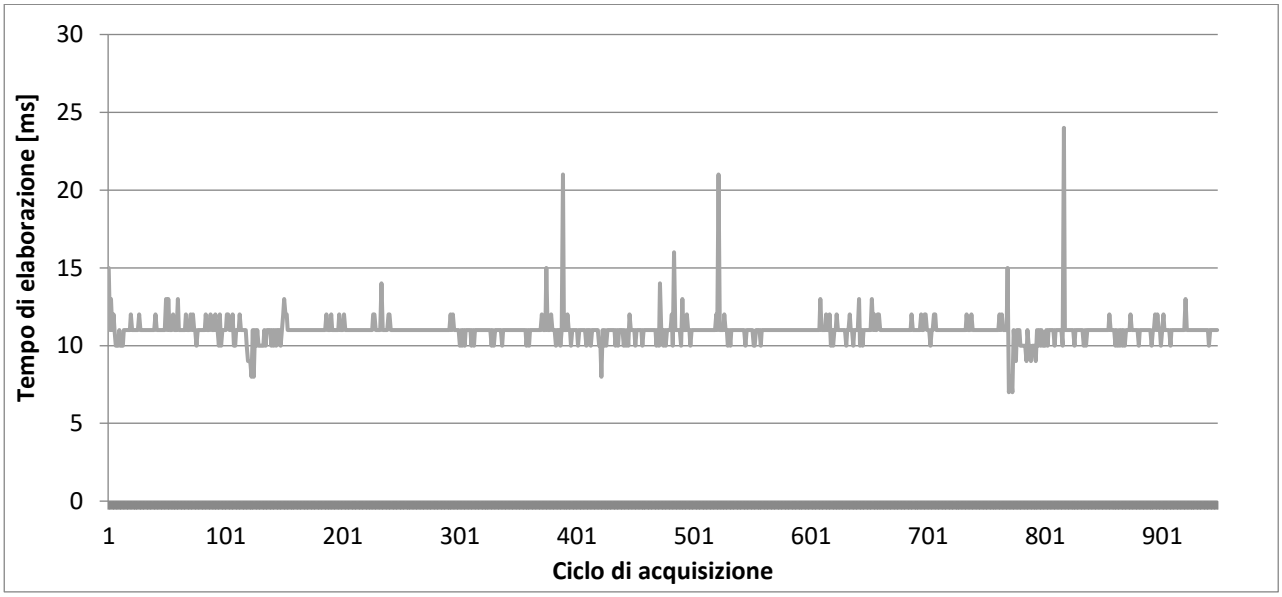


**Grafico 1. Tempi di elaborazione con GPU Tesla K40**

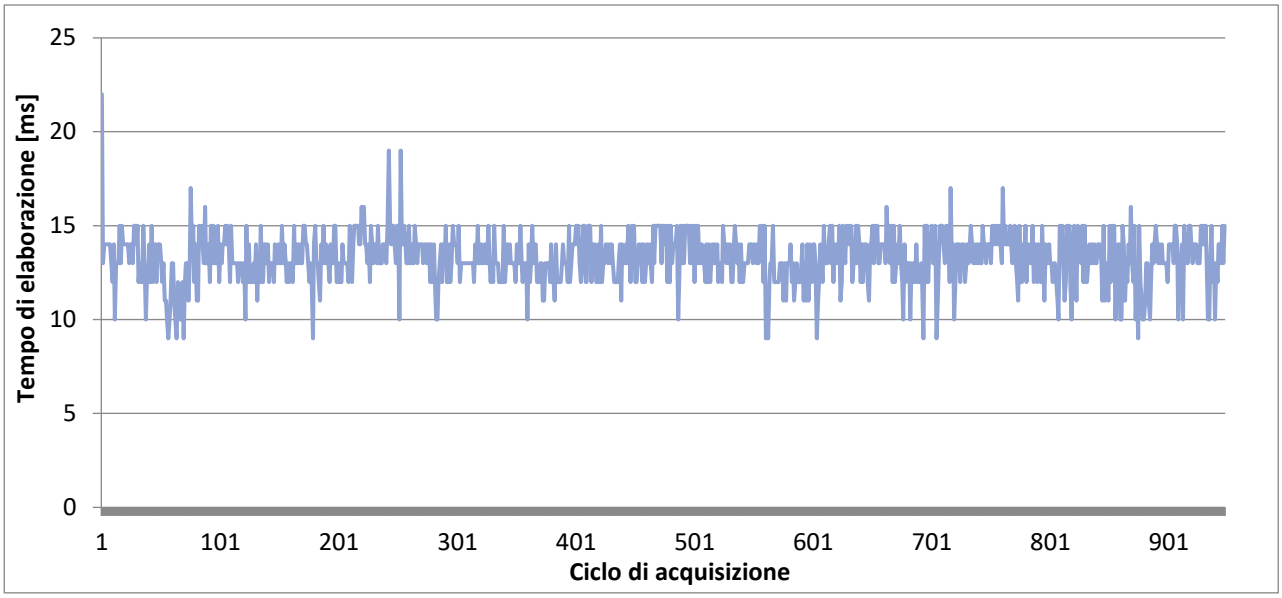


**Grafico 2. Tempi di elaborazione con GPU Geforce GTX Titan**

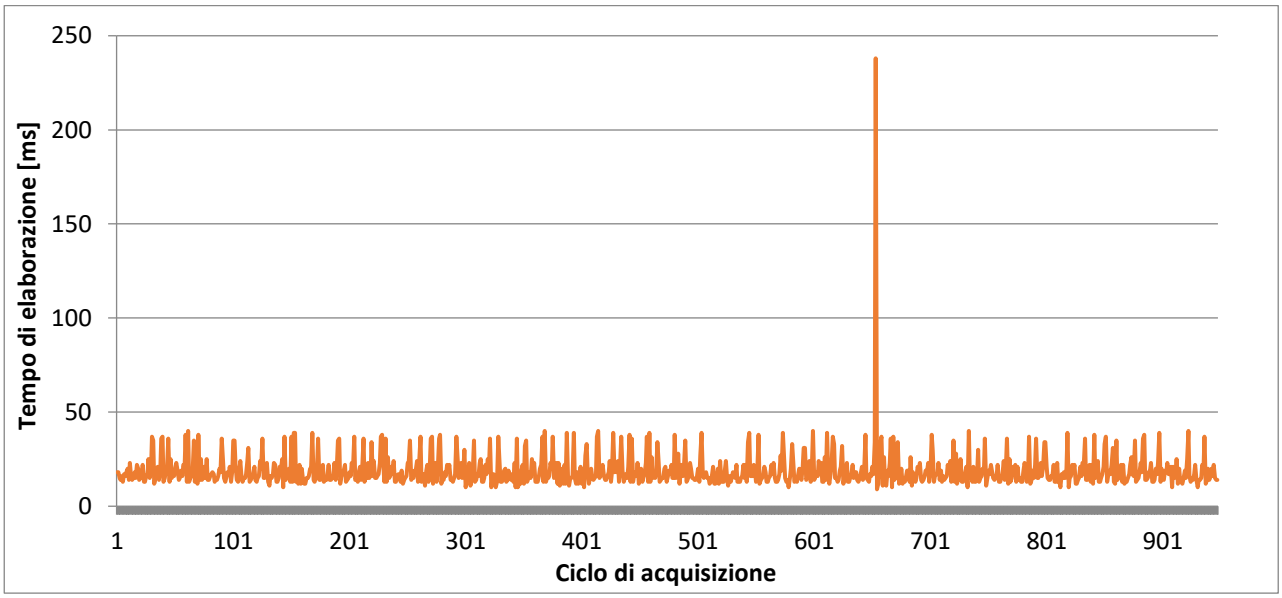


**Grafico 3. Tempi di elaborazione con CPU**

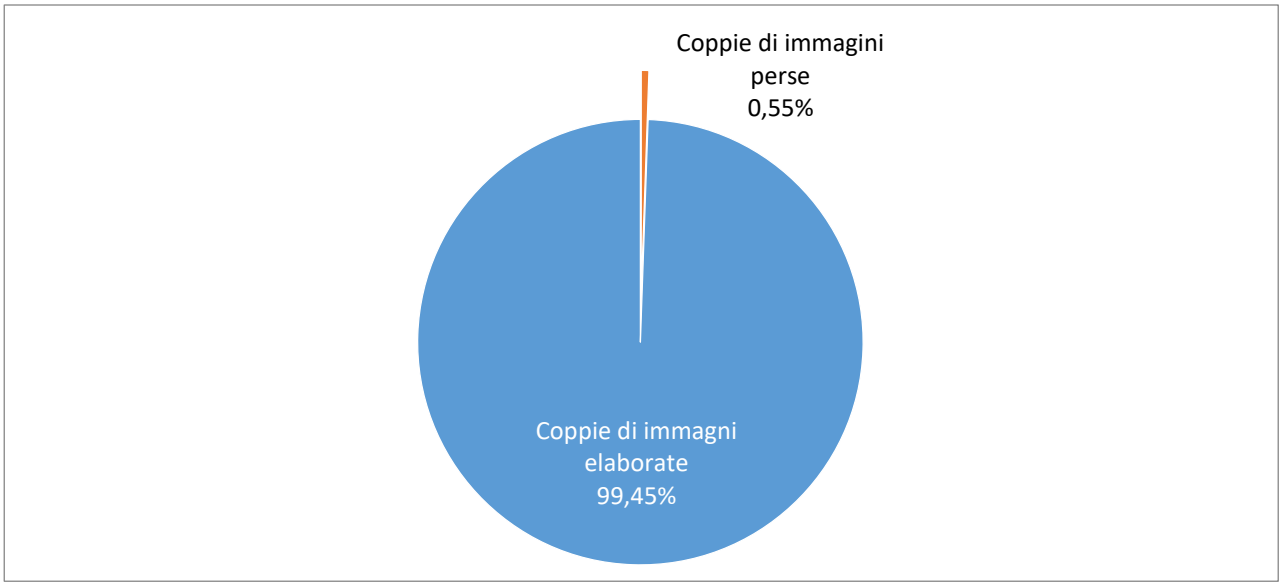


**Grafico 4. Percentuale delle immagini elaborate con GPU Tesla K40**

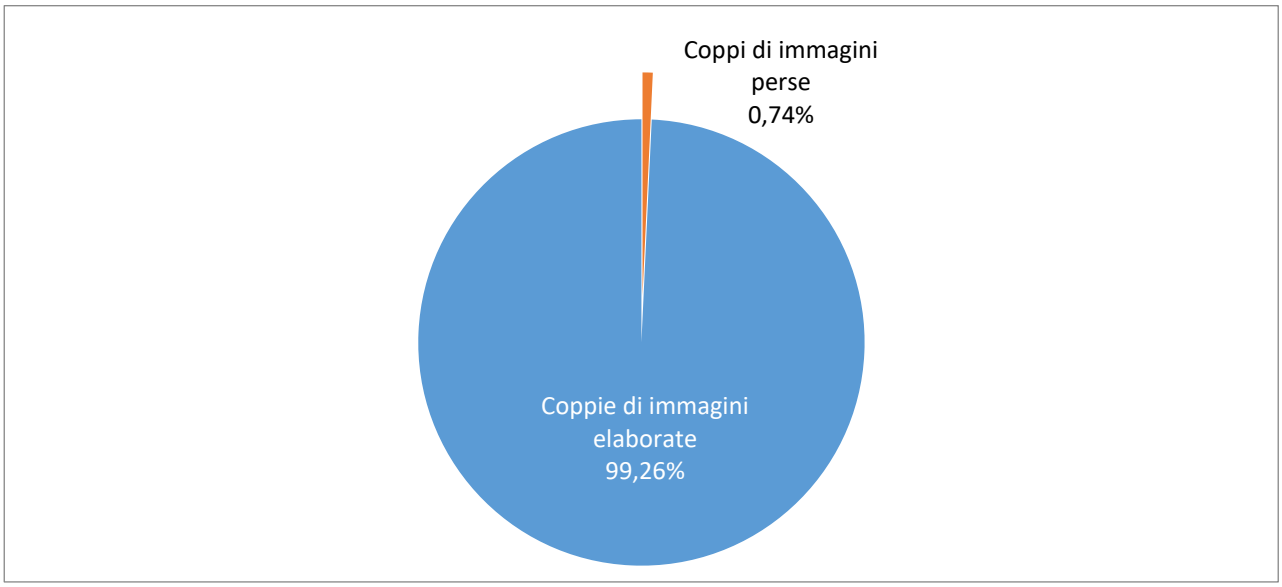


**Grafico 5. Percentuale delle immagini elaborate con GPU Geforce GTX Titan**

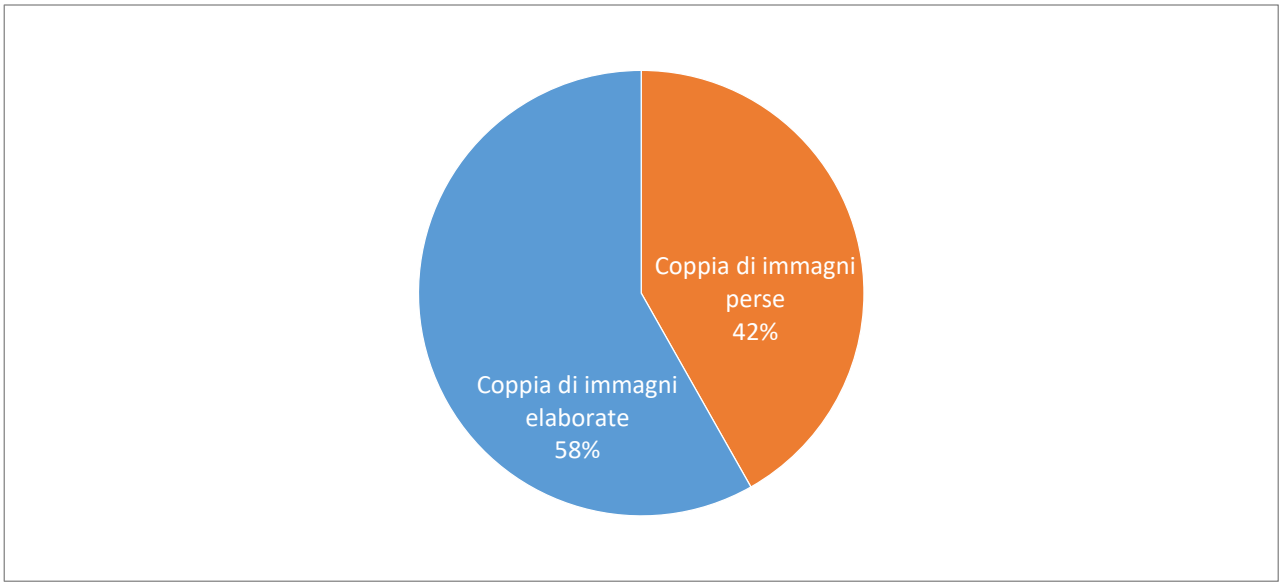


**Grafico 6. Percentuale delle immagini elaborate con CPU**

Come detto in precedenza, se l'elaborazione di una coppia di immagini richiede più tempo del periodo di acquisizione il supervisore evita di elaborare la coppia appena acquisita.

Come dimostrano i log di elaborazione allegati, la riduzione della percentuali di coppie di immagini non elaborate con l'utilizzo della GPU è notevole. Sfruttando le potenzialità di calcolo del dispositivo, la percentuale di coppie di immagini perse si riduce al 0,55% delle coppie acquisite, mentre con la CPU la percentuale è pari al 42%. Questo si traduce in una maggiore fluidità del movimento dell'indicatore sullo schermo.

Inoltre, paragonando le prestazioni delle due GPU, la Tesla K40 risulta più performante della sua concorrente confermando di essere un dispositivo ideale per il calcolo. Tutte le considerazioni che seguono si riferiscono a questo modello.

# 4. Gestione della memoria per la condivisione dati PC-GPU

Gli algoritmi di elaborazioni delle immagini stereo possono essere eseguiti su GPU impiegando in media l' 80% del tempo che richiederebbe l'esecuzione su CPU. In particolare, è possibile ottimizzare su GPU l'operazione di remapping e sogliatura delle immagini, mentre la ricerca dei marker e la ricostruzione 3D possono essere eseguite esclusivamente su CPU.

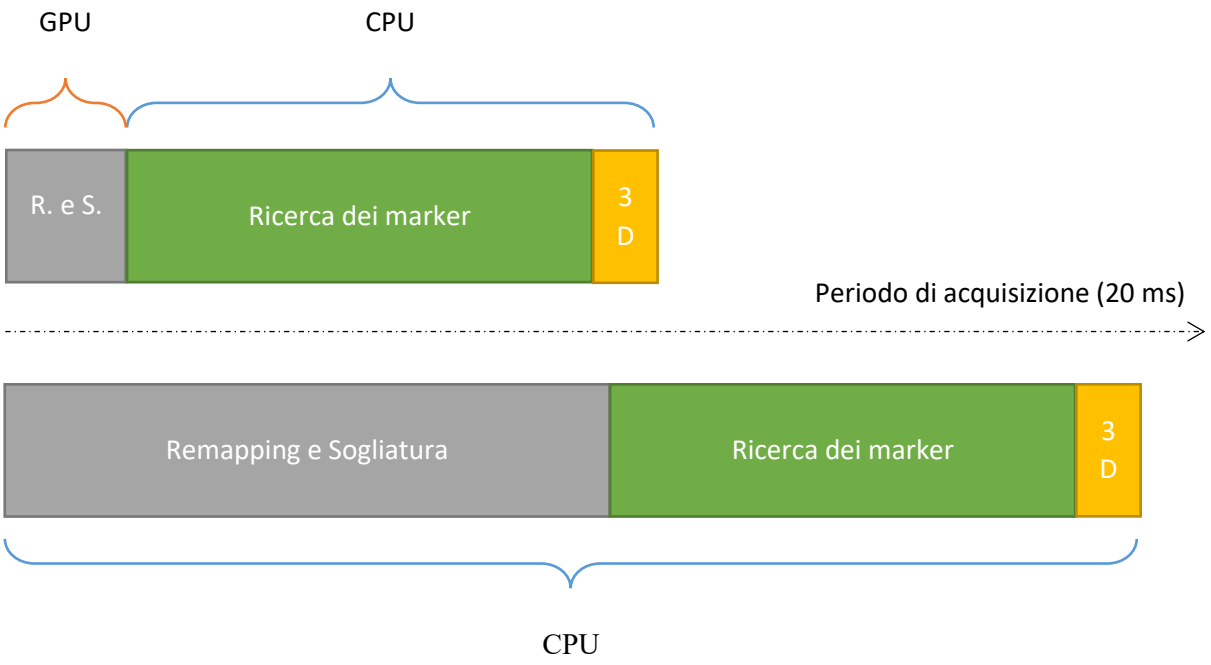


**Figura 4. Confronto dei tempi di elaborazione delle coppie di immagini**

Per sfruttare al meglio il guadagno sull'elaborazione è opportuno che le immagini siano trasferite sulla memoria dell'unità di elaborazione in modo efficiente. Il tempo necessario per caricare e scaricare i dati dal dispositivo deve essere considerato quando si valuta la portabilità di un'applicazione su GPU, poiché potrebbe incidere notevolmente sul tempo di esecuzione complessivo.

Le strategie disponibili per trasferire i dati da elaborare sulla GPU sono molteplici con diversi impatti sulle prestazioni. Sono state valutate tre strategie.

## 4.1 Utilizzo standard della comunicazione memoria PC-memoria GPU

Le allocazioni di memoria RAM sono paginabili; l'accesso ad una pagina di memoria non disponibile in memoria centrale genera un page fault, di conseguenza è necessario che il sistema operativo recuperi la pagina richiesta dalla memoria virtuale (richiedendo calcolo di CPU) prima di trasferirla al dispositivo. Per questo motivo, operazioni di upload e download di dati verso il dispositivo sono operazioni sincrone [2].

## 4.2 Definizione stream di elaborazione su GPU

Onde evitare che l'accesso ad una pagina di memoria generi un page fault, è possibile allocare in memoria RAM uno spazio non paginabile (page-locked o pinned). Quest'area di memoria sarà sempre presente nella memoria centrale, per cui il trasferimento da e verso questa allocazione può essere gestito dal DMA senza coinvolgere la CPU, diventano quindi operazioni asincrone [3].

L'allocazione di memoria non paginabile consente l'utilizzo degli stream, ossia sequenze di operazioni eseguite sulla GPU nell'ordine con cui vengono definite nel codice. Mentre le operazioni all'interno di uno stream sono eseguite in un ordine prestabilito, le operazioni di differenti stream possono essere intervallate e, quando possibile, persino eseguite contemporaneamente [4].

La memoria non paginabile è una risorsa preziosa in quanto allocando memoria non paginabile si riduce la quantità di memoria fisica disponibile per il sistema operativo e per le altri applicazioni. Questo potrebbe avere ripercussioni sulle prestazioni dell'intero sistema.

## 4.3 Utilizzo del mapping tra memoria del PC e spazio degli indirizzi della GPU

Le allocazioni di memoria non paginabili che contengono i dati da elaborare possono non essere trasferite sul dispositivo utilizzando il meccanismo della “zero copy”: i thead in esecuzione sulla GPU hanno accesso diretto alla memoria RAM avvalendosi di un meccanismo di mapping dello spazio degli indirizzi della CPU e della GPU, offrendo un vero e proprio meccanismo di condivisione e comunicazione [5]. Questa scelta è conveniente quando la GPU è integrata nella scheda madre in quanto evita inutili copie di dati considerando che la memoria della GPU integrata è la stessa della CPU. Nelle GPU esterne questa scelta può essere vantaggiosa se i dati in memoria sono letti e scritti solo una volta, poiché non possono essere memorizzati nella memoria cache della GPU.

Questa strategia può essere utilizzata al posto degli stream poiché le operazioni di trasferimento dati sono sovrapposte alle operazioni di calcolo (si risparmia quindi sul tempo necessario per il set up degli stream).

## 4.4 Impatto sulle performance e confronto dei risultati

Utilizzando la GPU per l’elaborazione delle immagini e gestendo la memoria non imponendo alcun vincolo sulla paginazione si ottiene un tempo di elaborazione medio pari a 10,94 ms e deviazione standard pari a 0,87 ms. Grazie alla notevole disponibilità di banda offerta dal collegamento PCIe, la copia dei dati in maniera sincrona non incide negativamente sulle prestazioni.

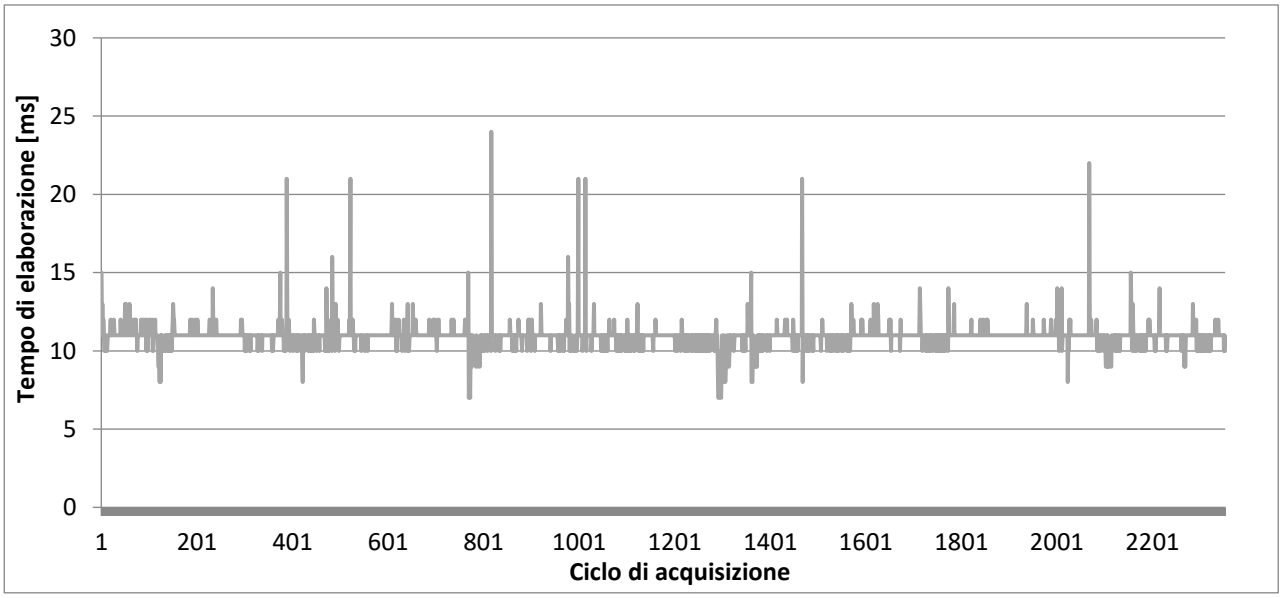


**Grafico 7. Tempi di elaborazione con GPU e uso della memoria paginabile**

Sfruttando l'elaborazione asincrona attraverso gli stream e allocando memoria non paginabile si paga un costo iniziale di inizializzazione e si ottiene un tempo di elaborazione medio pari a 11,07 ms e deviazione standard pari a 0.78 ms. Poiché la CPU non può effettuare altri calcoli prima che la GPU termini la propria elaborazione, questa strategia non offre miglioramenti in termini di prestazioni ma produce un miglior utilizzo delle risorse di GPU in quanto le elaborazioni delle due immagini avvengono in maniera concorrente.

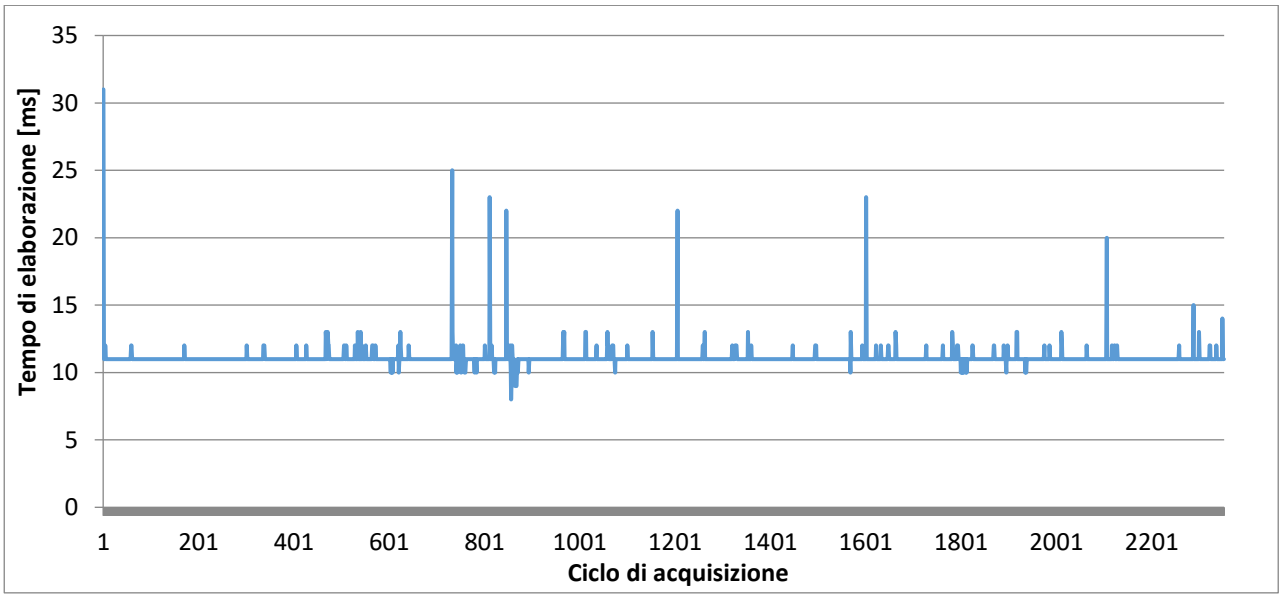


Grafico 8. Tempi di elaborazione con GPU e uso della memoria "page-locked"

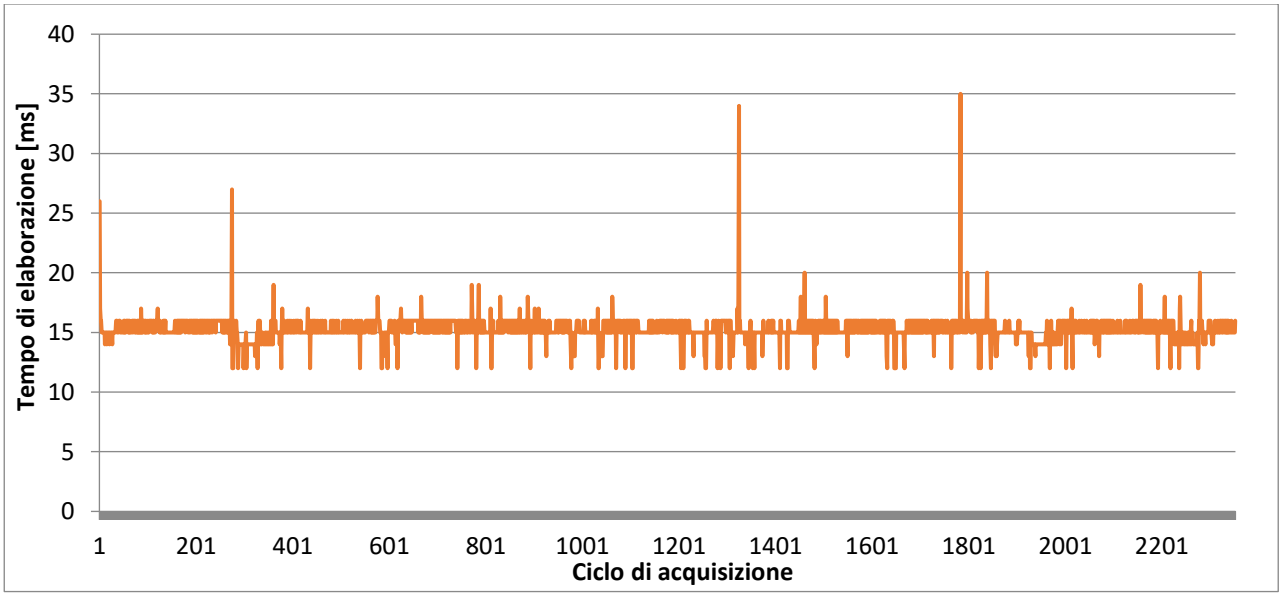


Grafico 9. Tempi di elaborazione con GPU e uso della memoria "zero copy"

# 5. Ottimizzazione dell'elaborazione GPU

## 5.1 Algoritmi

Per quanto concerne gli algoritmi di elaborazione delle immagini si è utilizzata una libreria con funzioni ottimizzate per GPU. È stato necessario quindi valutare quali funzioni utilizzare e con quali parametri per ottenere il massimo delle prestazioni.

## 5.2 Gestioni della memoria

Verificati gli algoritmi ed i parametri, la gestione della memoria è stata discriminante e decisiva per l'ottimizzazione dell'elaborazione su GPU come mostrano i grafici riportati di seguito. Questi tengono conto solo del tempo di calcolo sul dispositivo trascurando le eventuali operazioni di sincronizzazione (necessarie con l'utilizzo degli stream).

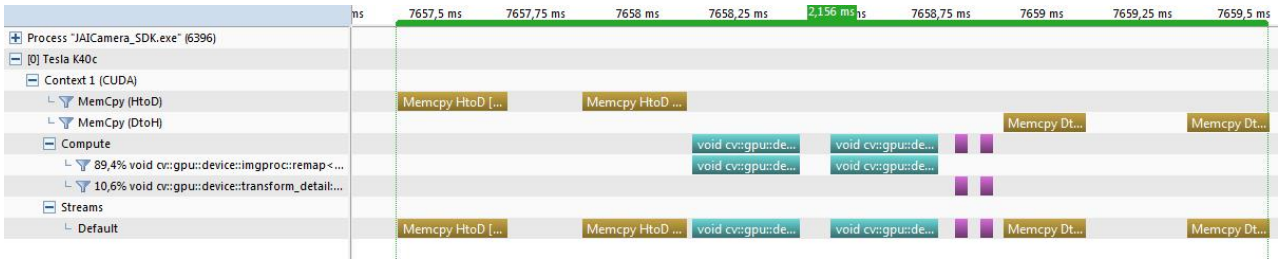


**Figura 5. Utilizzo della GPU con memoria paginabile**

L'utilizzo della memoria non paginabile non consente l'utilizzo degli stream per cui le operazioni sulla GPU vengono svolte in sequenza. Nel momento in cui le immagini sono trasferite sulla memoria del dispositivo l'accesso ai dati è estremamente veloce. Il 75% del tempo di esecuzione è impiegato per il solo trasferimento di dati.

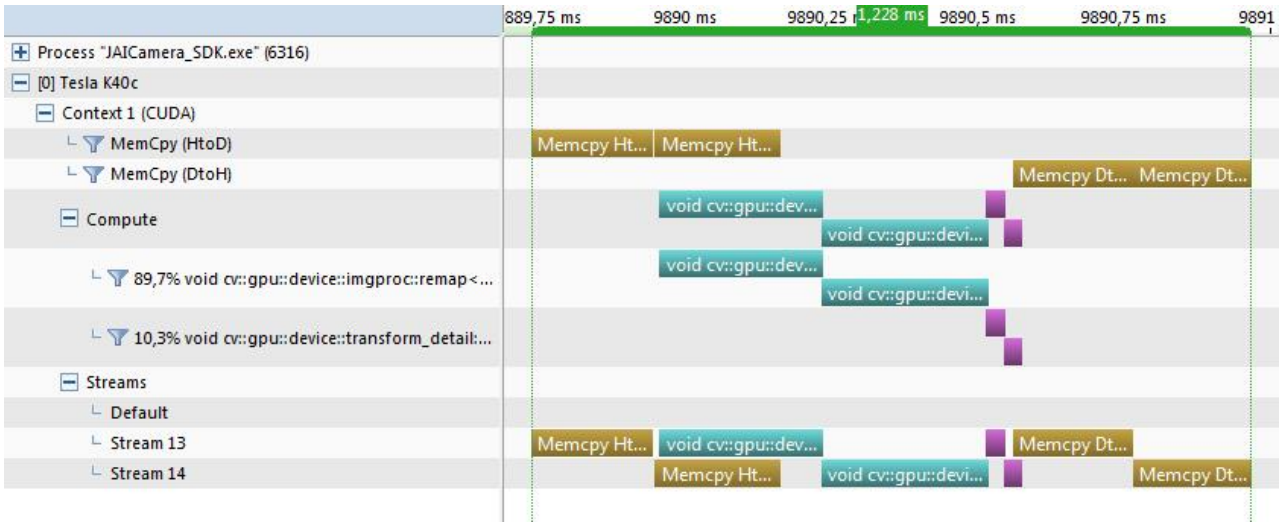


**Figura 6. Utilizzo della GPU con memoria "page-locked"**

L’utilizzo della memoria non paginabile e degli stream costituisce la soluzione ottimale poiché, nonostante la durata delle singole operazioni sia analoga al caso precedente, le operazioni di copia dei dati e di calcolo sono sovrapposte riducendo in media del 43% il tempo complessivo. Al tempo di calcolo su GPU va sommato il tempo necessario per la sincronizzazione degli stream.

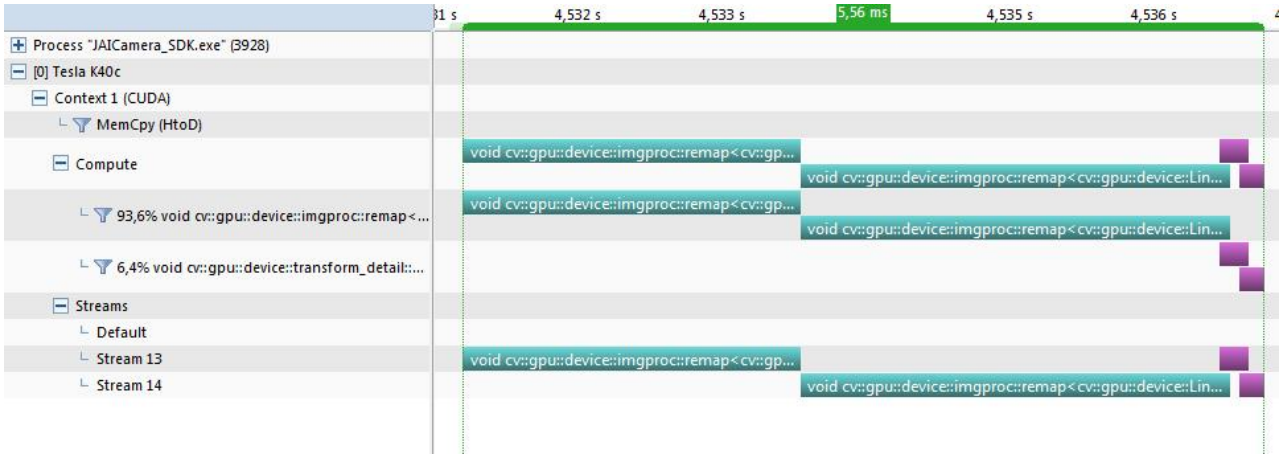


**Figura 7. Utilizzo della GPU con memoria “zero copy”**

Applicando la strategia della “zero copy” si evitano trasferimenti di memoria da e verso la GPU ma il tempo di esecuzione delle operazioni di calcolo aumenta significativamente poiché i dati di lettura e scrittura risiedono in memoria RAM.

In conclusione, l’utilizzo della memoria non paginabile costituisce la soluzione ottimale in termini di riduzione del tempo di elaborazione ed utilizzo delle risorse di calcolo offerte dalla GPU.

# Appendice 1 – Struttura del Log di elaborazione con CPU

Si mostra di seguito la struttura dei file Log prodotti relativamente alla elaborazione con CPU. I file completi, di 5154 righe non sono riportati in forma stampata per economia di carta, ma sono inclusi nel DVD allegato.

I log riportano (in un ordine che dipende dall'elaborazione):

- La coppia di immagini acquisite con relativi timestamp
- Il tempo impiegato dalla CPU per l'elaborazione delle immagini e la ricerca dei marker
- Il tempo complessivo impiegato per l'elaborazione della coppia di frame acquisite (elaborazione delle immagini, ricostruzione 3D, operazioni di memoria)
- La coppia di immagini elaborate con relativi timestamp
- Coppie di immagini non elaborate (Watcher busy)

Acquired 1 ( 6744032 , 6017680 )

Acquired 2 ( 26750128 , 26023952 )

CPU Pre-processing + Marker detection [ms]: 18

Elapsed time [ms]: 18

Processed 1 ( 6744032 , 6017680 )

Acquired 3 ( 46742464 , 46016496 )

CPU Pre-processing + Marker detection [ms]: 18

Elapsed time [ms]: 18

Processed 2 ( 26750128 , 26023952 )

Acquired 4 ( 66748560 , 66022784 )

CPU Pre-processing + Marker detection [ms]: 14

Elapsed time [ms]: 15

Processed 3 ( 46742464 , 46016496 )

CPU Pre-processing + Marker detection [ms]: 13

Elapsed time [ms]: 14

Processed 4 ( 66748560 , 66022784 )

Acquired 5 ( 86740896 , 86015312 )

# Appendice 2 – Struttura del Log di elaborazione con GPU NVIDIA Tesla K40

Si mostra di seguito la struttura dei file Log prodotti relativamente alla elaborazione con GPU NVIDIA Tesla K40 e utilizzo della memoria "page locked". I file completi, di 25754 righe non sono riportati in forma stampata per economia di carta, ma sono inclusi nel DVD allegato.

I log riportano (in un ordine che dipende dall'elaborazione):

- La coppia di immagini acquisite con relativi timestamp
- Il tempo impiegato dalla GPU per l'elaborazione delle immagini
- Il tempo impiegato dalla CPU per la ricerca dei marker
- Il tempo complessivo impiegato per l'elaborazione della coppia di frame acquisite (elaborazione delle immagini, ricostruzione 3D, operazioni di memoria)
- La coppia di immagini elaborate con relativi timestamp
- Coppie di immagini non elaborate (Watcher busy)

Acquired 1 ( 7133600 , 6259424 )

GPU [ms] 2

Acquired 2 ( 27125952 , 26251952 )

Watcher busy

CPU Marker detection [ms]: 27

Elapsed time [ms]: 31

Processed 1 ( 7133600 , 6259424 )

Acquired 3 ( 47132048 , 46258240 )

GPU [ms] 2

CPU Marker detection [ms]: 8

Elapsed time [ms]: 11

Processed 2 ( 47132048 , 46258240 )

Acquired 4 ( 67124384 , 66250784 )

GPU [ms] 2

CPU Marker detection [ms]: 8

Elapsed time [ms]: 11

Processed 3 ( 67124384 , 66250784 )

Acquired 5 ( 87130480 , 86257072 )

GPU [ms] 2

CPU Marker detection [ms]: 8

Elapsed time [ms]: 11

## Appendice 3 – Struttura del Log di elaborazione con GPU NVIDIA Geforce Titan

Si mostra di seguito la struttura dei file Log prodotti relativamente alla elaborazione con GPU NVIDIA Geforce Titan. I file completi, di 14933 righe non sono riportati in forma stampata per economia di carta, ma sono inclusi nel DVD allegato.

I log riportano (in un ordine che dipende dall'elaborazione):

- La coppia di immagini acquisite con relativi timestamp
- Il tempo impiegato dalla GPU per l'elaborazione delle immagini
- Il tempo impiegato dalla CPU per la ricerca dei marker
- Il tempo complessivo impiegato per l'elaborazione della coppia di frame acquisite (elaborazione delle immagini, ricostruzione 3D, operazioni di memoria)
- La coppia di immagini elaborate con relativi timestamp
- Coppie di immagini non elaborate (Watcher busy)

Acquired 1 ( 6758320 , 6320432 )

GPU [ms] 11

Acquired 2 ( 26764416 , 26326720 )

Watcher busy

CPU Marker detection [ms]: 10

Elapsed time [ms]: 22

Processed 1 ( 6758320 , 6320432 )

Acquired 3 ( 46756752 , 46319248 )

GPU [ms] 4

CPU Marker detection [ms]: 8

Elapsed time [ms]: 13

Processed 2 ( 46756752 , 46319248 )

Acquired 4 ( 66762832 , 66325520 )

GPU [ms] 4

CPU Marker detection [ms]: 8

Elapsed time [ms]: 14

Processed 3 ( 66762832 , 66325520 )

Acquired 5 ( 86755168 , 86318048 )

GPU [ms] 5

CPU Marker detection [ms]: 8

Elapsed time [ms]: 14

Processed 4 ( 86755168 , 86318048 )

# Appendice 4 – Struttura del Log di elaborazione con GPU NVIDIA Tesla K40 e memoria paginabile

Si mostra di seguito la struttura dei file Log prodotti relativamente alla elaborazione con GPU NVIDIA Tesla K40 ed utilizzo della memoria paginabile. I file completi, di 11791 righe non sono riportati in forma stampata per economia di carta, ma sono inclusi nel DVD allegato.

I log riportano (in un ordine che dipende dall'elaborazione):

- La coppia di immagini acquisite con relativi timestamp
- Il tempo impiegato dalla GPU per l'elaborazione delle immagini
- Il tempo impiegato dalla CPU per la ricerca dei marker
- Il tempo complessivo impiegato per l'elaborazione della coppia di frame acquisite (elaborazione delle immagini, ricostruzione 3D, operazioni di memoria)
- La coppia di immagini elaborate con relativi timestamp
- Coppie di immagini non elaborate (Watcher busy)

Acquired 1 ( 6747264 , 6071936 )

GPU [ms] 5

CPU Marker detection [ms]: 10

Elapsed time [ms]: 15

Processed 1 ( 6747264 , 6071936 )

Acquired 2 ( 26739600 , 26064464 )

GPU [ms] 2

CPU Marker detection [ms]: 8

Elapsed time [ms]: 11

Processed 2 ( 26739600 , 26064464 )

Acquired 3 ( 46745696 , 46070752 )

GPU [ms] 2

CPU Marker detection [ms]: 9

Elapsed time [ms]: 13

Processed 3 ( 46745696 , 46070752 )

Acquired 4 ( 66751792 , 66063296 )

GPU [ms] 2

CPU Marker detection [ms]: 8

Elapsed time [ms]: 11

Processed 4 ( 66751792 , 66063296 )

Acquired 5 ( 86744144 , 86069584 )

GPU [ms] 2

CPU Marker detection [ms]: 8

Elapsed time [ms]: 12

Processed 5 ( 86744144 , 86069584 )

## Appendice 5 – Struttura del Log di elaborazione con GPU NVIDIA Tesla K40 e “zero copy memory”

Si mostra di seguito la struttura dei file Log prodotti relativamente alla elaborazione con GPU NVIDIA Tesla K40 ed utilizzo della “zero copy”. I file completi, di 16154 righe non sono riportati in forma stampata per economia di carta, ma sono inclusi nel DVD allegato.

I log riportano (in un ordine che dipende dall’elaborazione):

- La coppia di immagini acquisite con relativi timestamp
- Il tempo impiegato dalla GPU per l’elaborazione delle immagini
- Il tempo impiegato dalla CPU per la ricerca dei marker
- Il tempo complessivo impiegato per l’elaborazione della coppia di frame acquisite (elaborazione delle immagini, ricostruzione 3D, operazioni di memoria)
- La coppia di immagini elaborate con relativi timestamp
- Coppie di immagini non elaborate (Watcher busy)

Acquired 1 ( 6729008 , 5977472 )

GPU [ms] 7

Acquired 2 ( 26735104 , 25970016 )

Watcher busy

CPU Marker detection [ms]: 10

Elapsed time [ms]: 26

Processed 1 ( 6729008 , 5977472 )

Acquired 3 ( 46727456 , 45976288 )

GPU [ms] 7

Acquired 4 ( 66733552 , 65968832 )

CPU Marker detection [ms]: 9

Elapsed time [ms]: 17

Processed 2 ( 46727456 , 45976288 )

GPU [ms] 7

Acquired 5 ( 86725904 , 85975120 )

CPU Marker detection [ms]: 8

Elapsed time [ms]: 16

Processed 3 ( 66733552 , 65968832 )

GPU [ms] 6

Acquired 6 ( 106732000 , 105967664 )

CPU Marker detection [ms]: 8

Elapsed time [ms]: 16

Processed 4 ( 86725904 , 85975120 )

## Appendice 6 – Video elaborati

Nel DVD sono presenti video registrati per lo sviluppo e il test degli algoritmo. Non sono naturalmente quelli che hanno generato i log di cui agli appendici 1-6 poiché nella determinazione dei log si è utilizzato come input video live escludendo la procedura di salvataggio per non penalizzare le prestazioni.

## Appendice 7 – Codice sorgente

Nel DVD è inoltre presente una cartella con il codice sorgente del progetto software sviluppato in C++ con framework Qt 5.2 e CUDA 7.5.

## Bibliografia


[1] HTTPS://WWW.TOP500.ORG/LISTS/2016/06/

[2] HTTP://DOCS.NVIDIA.COM/CUDA/CUDA-C-PROGRAMMING-GUIDE/#DEVICE-MEMORY

[3] HTTP://DOCS.NVIDIA.COM/CUDA/CUDA-C-PROGRAMMING-GUIDE/#PAGE-LOCKED-HOST-MEMORY

[4] J. CHENG, M. GROSSMAN, T. MCKERCHER, "PROFESSIONAL CUDA C PROGRAMMING", JOHN WILEY & SONS, 2014

[5] HTTP://DOCS.NVIDIA.COM/CUDA/CUDA-C-PROGRAMMING-GUIDE/#MAPPED-MEMORY