



Tesi di laurea triennale in informatica

ANZALDI LUCA

# Accelerazione di algoritmi di Stereo Matching su GPU per sistemi SLAM



# I sistemi ORB - SLAM

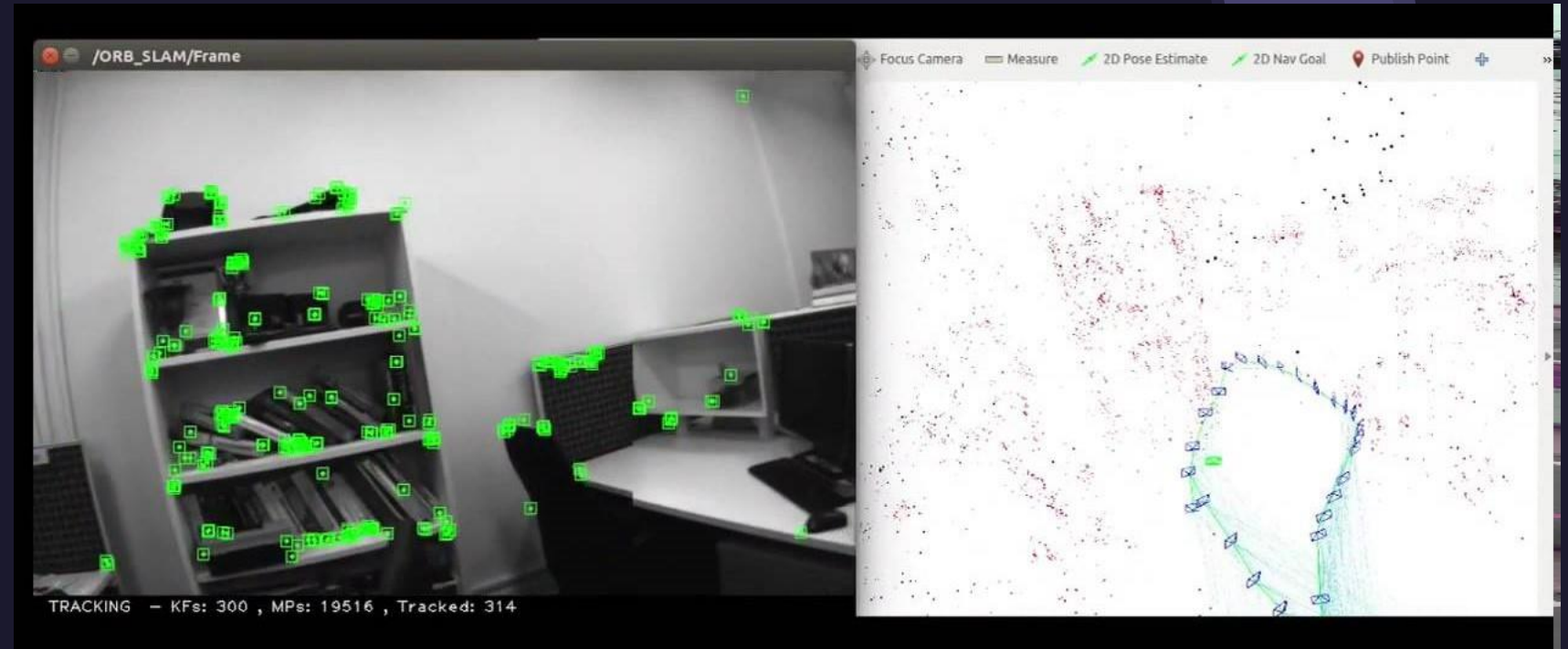
# Sistemi ORB - SLAM

- Caratteristiche principali ↩
- Campi di utilizzo
- Analisi della funzione di Stereo Matching

Utilizza algoritmo **ORB**  
(Oriented FAST and  
Rotated BRIEF)

## ! STEREO MATCHING

### 1. Rilevamento di caratteristiche visive

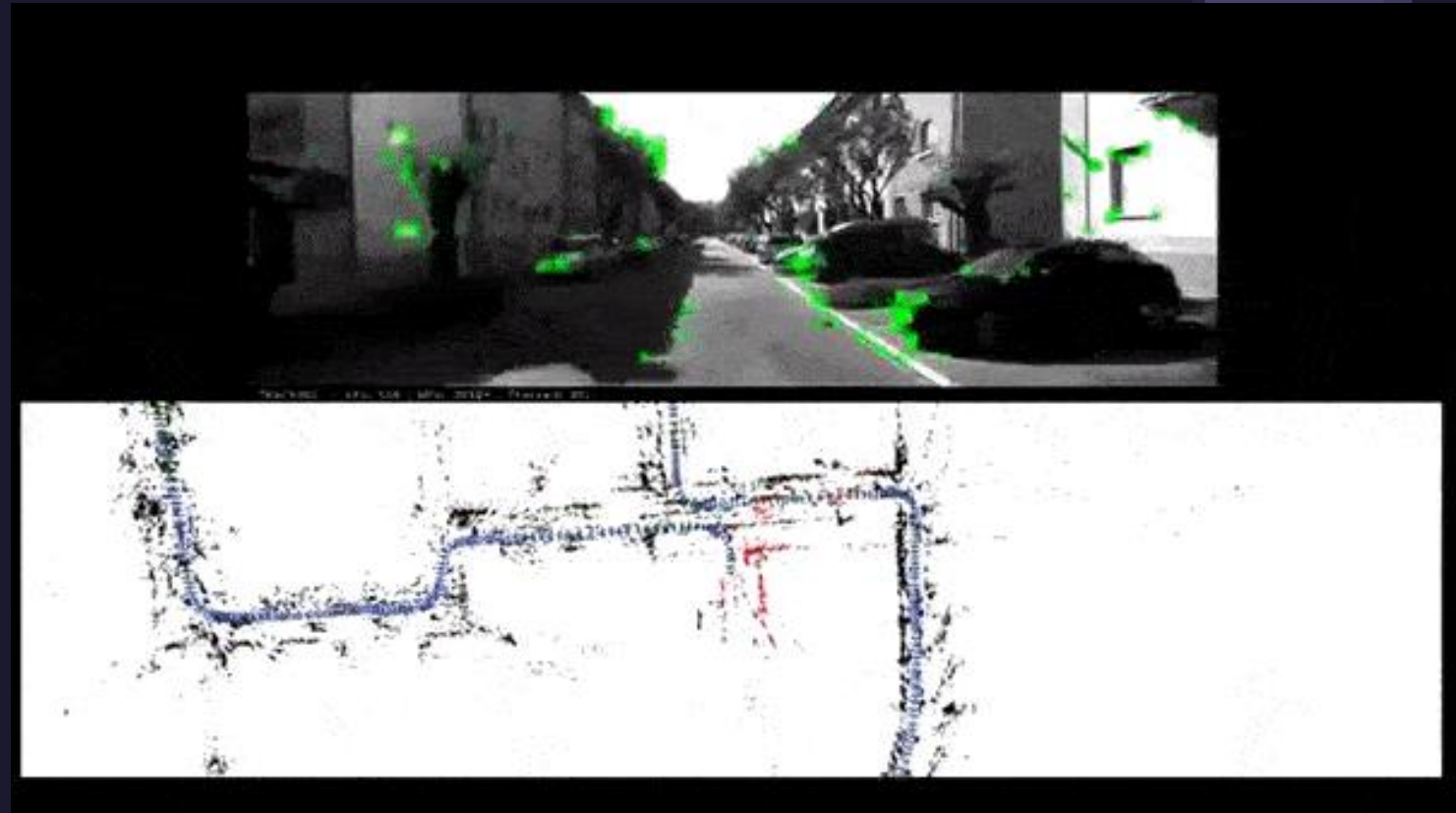




# Sistemi ORB - SLAM

- Caratteristiche principali ↩
- Campi di utilizzo
- Analisi della funzione di Stereo Matching

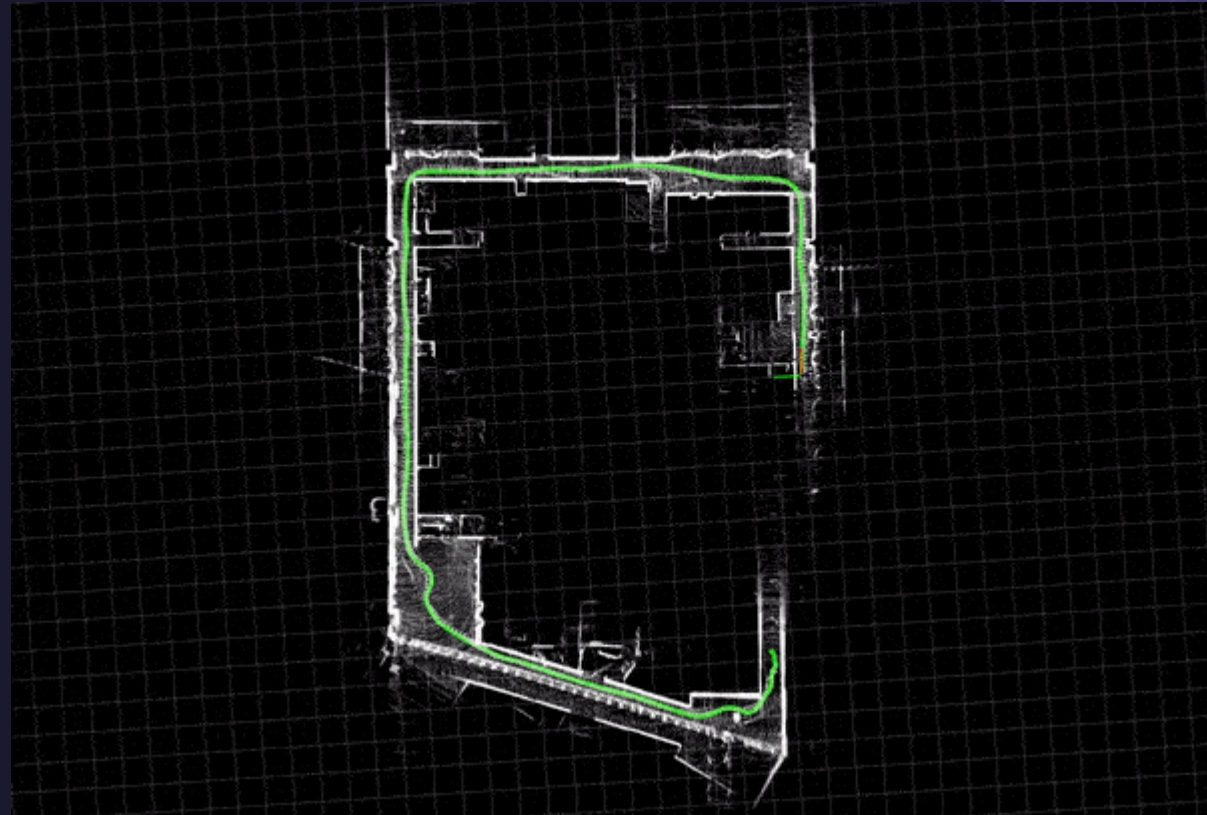
## 2. Mappatura



# Sistemi ORB - SLAM

- Caratteristiche principali ↩
- Campi di utilizzo
- Analisi della funzione di Stereo Matching

## 3. Localizzazione e riconoscimento dei loop



# Sistemi ORB - SLAM

- Caratteristiche principali
- Campi di utilizzo ↩
- Analisi della funzione di Stereo Matching



Veicoli Autonomi



Realtà aumentata



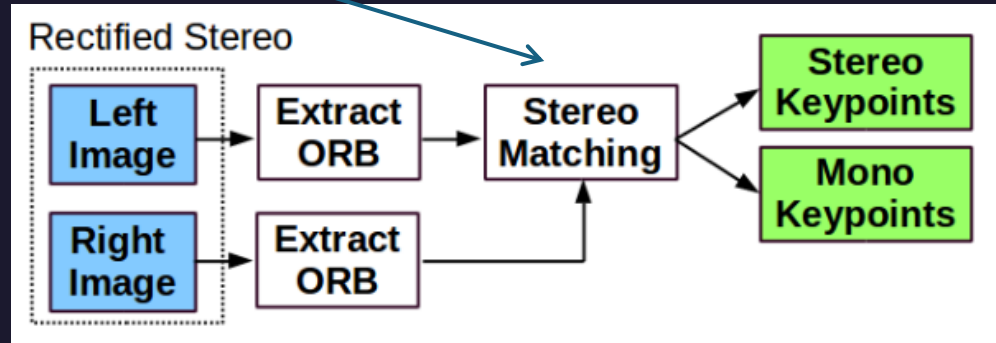
Droni



Robot

# Sistemi ORB - SLAM

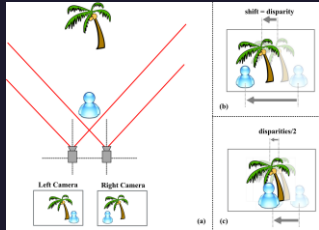
- Caratteristiche principali
- Campi di utilizzo
- Analisi della funzione di Stereo Matching ←



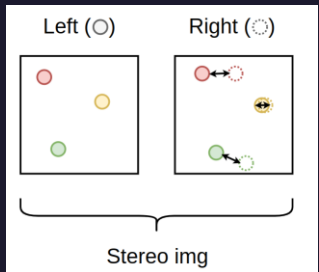
# Sistemi ORB – SLAM – Stereo Matching



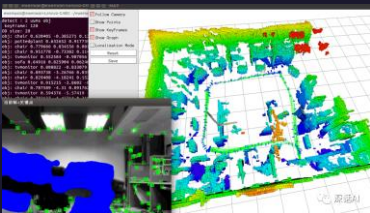
Acquisizione delle immagini



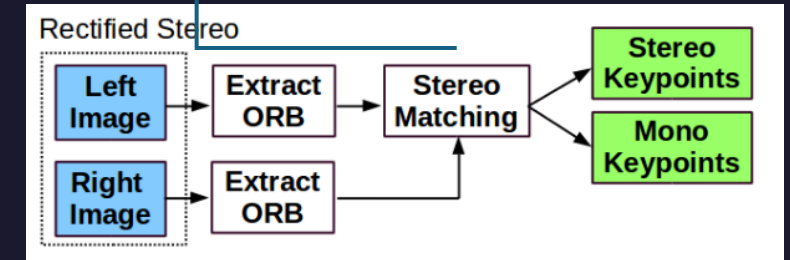
① Corrispondenza dei pixel



② Disparità



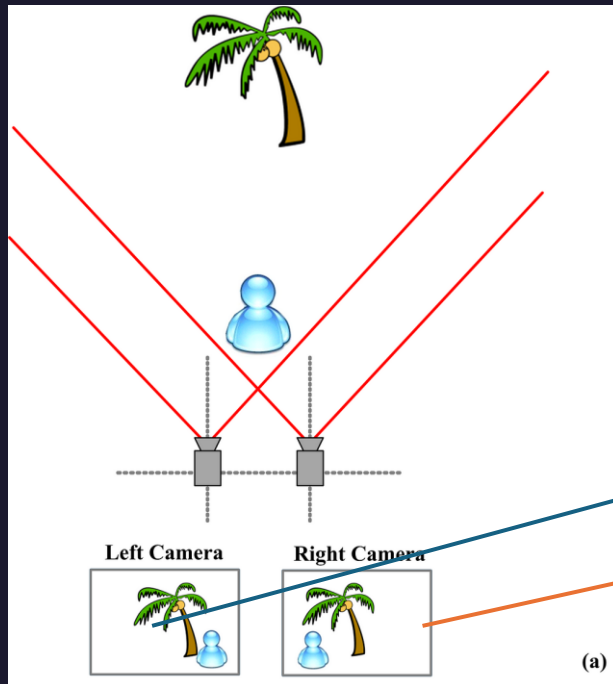
Ricostruzione 3D



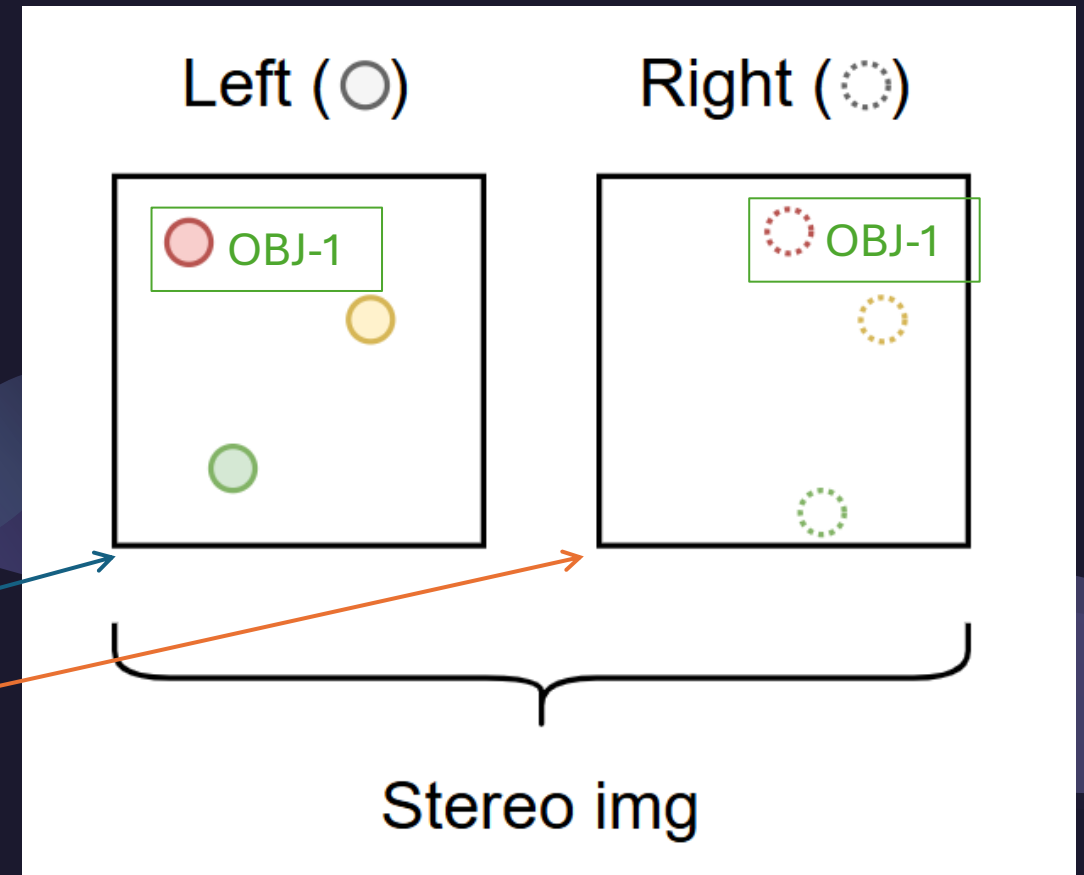


# Sistemi ORB – SLAM - Stereo Matching

- Caratteristiche principali
- Campi di utilizzo
- Analisi della funzione di Stereo Matching ←

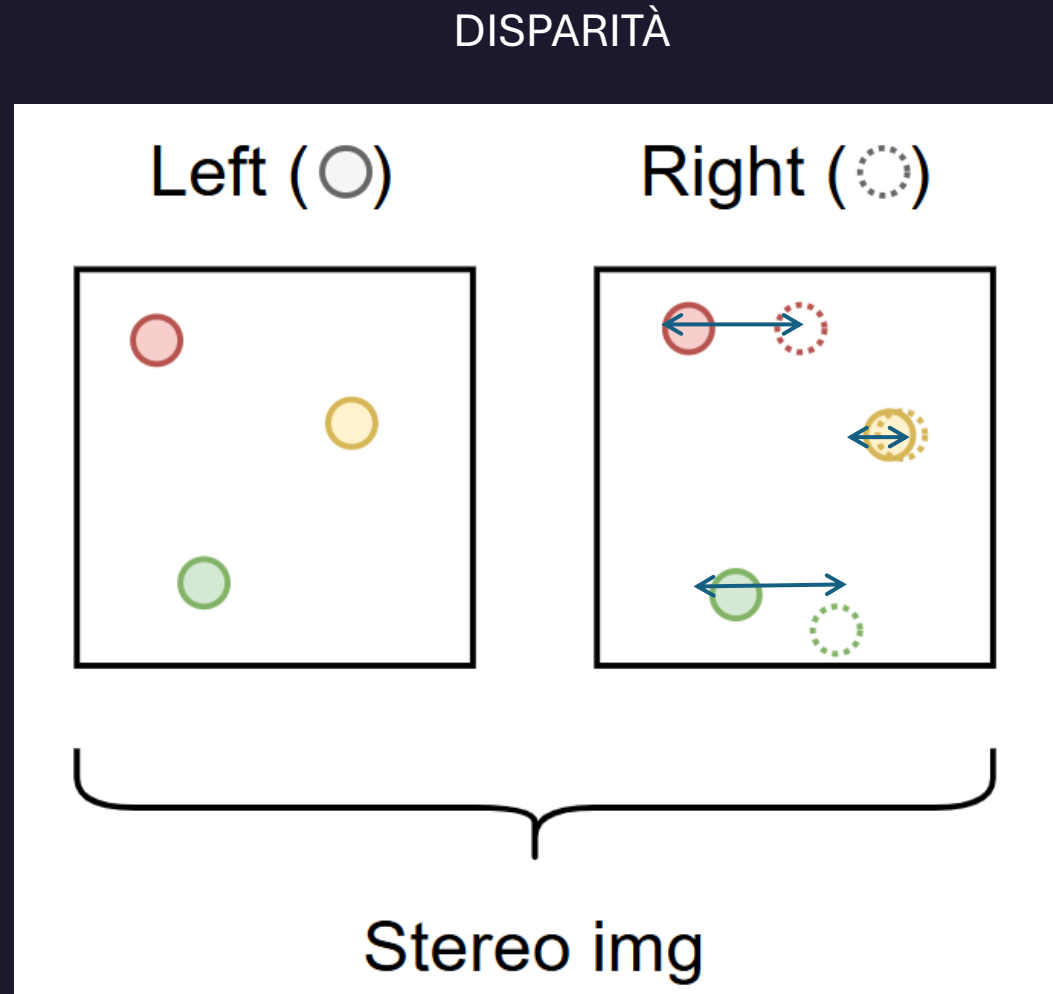


CORRISPONDENZA DEI PIXEL



# Sistemi ORB - ~~SILAMI~~ - Stereo Matching

- Caratteristiche principali
- Campi di utilizzo
- Analisi della funzione di Stereo Matching ↩





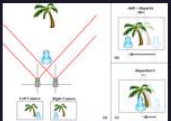
# Lavoro svolto per l'accelerazione dell'algoritmo

# Accelerazione algoritmo di Stereo Matching

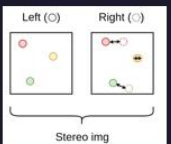
- Motivazioni accelerazione dell'algoritmo ↩
- Procedimento
- Risultati



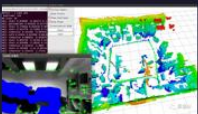
1 Acquisizione delle immagini



2 Corrispondenza dei pixel

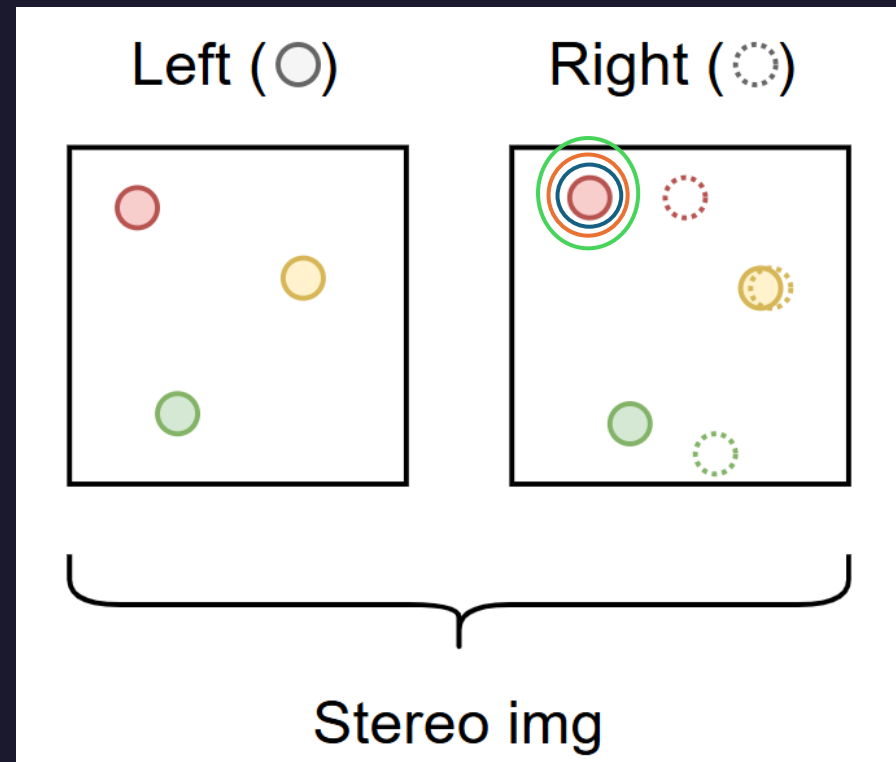


3 Disparità



4 Ricostruzione 3D

Corrispondenza dei pixel



○ Raggio = 1

○ Raggio = 2

○ Raggio = 4

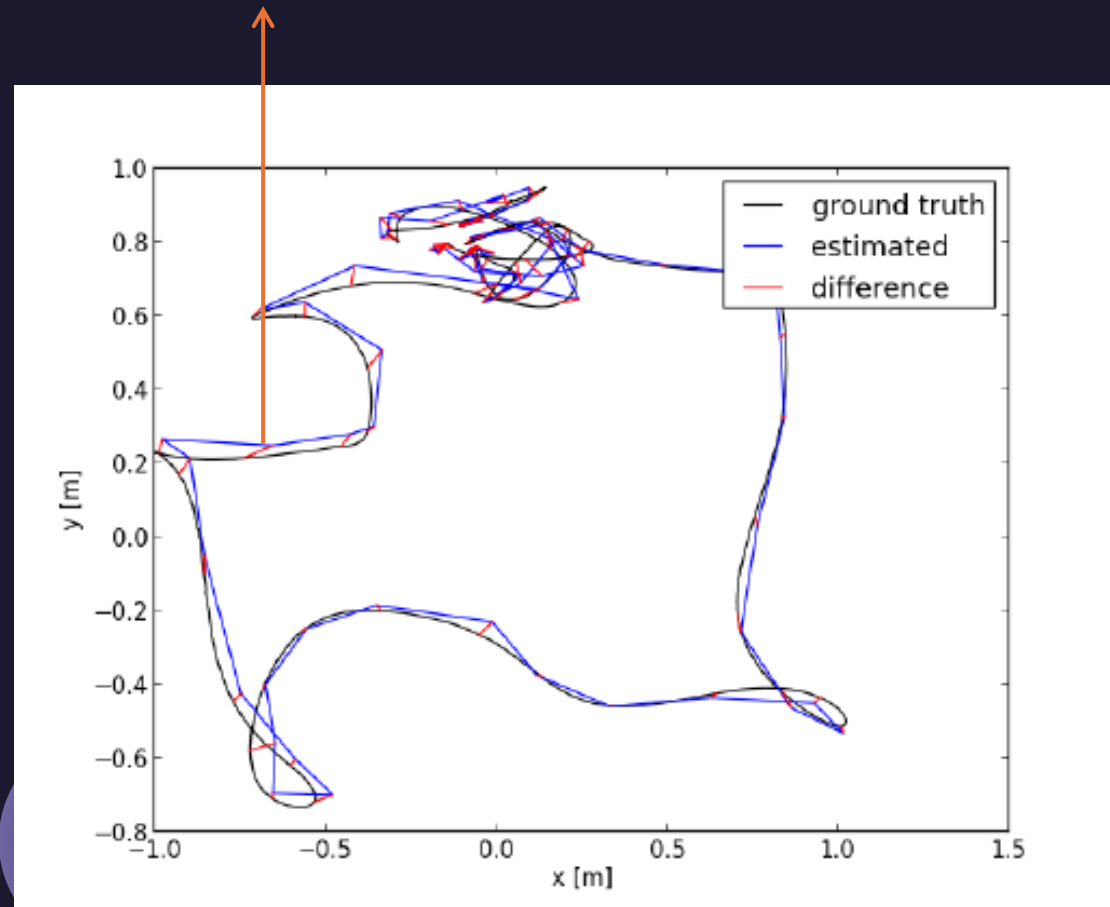
○ ...

# Accelerazione algoritmo di Stereo Matching

- Motivazioni accelerazione dell'algoritmo ↩
- Procedimento
- Risultati

$$ATE = \frac{1}{N} \sum_{i=1}^N \| \underline{T_{est}(i)} - \underline{T_{true}(i)} \|$$

ATE = Absolute Trajectory Error





# Accelerazione algoritmo di Stereo Matching

- Motivazioni accelerazione dell'algoritmo ←
- Procedimento
- Risultati

DATASET - 04

Radius multiplier	1.0	2.0	4.0	8.0	16.0
Translation error	0.47	0.46	0.46	0.47	0.47
Rotational error	0.07	0.08	0.09	0.07	0.07
ATE (m)	0.73	0.77	0.75	0.78	0.71
RPE (m)	0.017	0.017	0.017	0.017	0.017
RPE (deg)	0.031	0.032	0.031	0.030	0.030
ORB Extraction (ms)	9.1419	8.9207	8.9707	9.0029	8.9305
Stereo Matching (ms)	3.0216	3.391	3.962	4.8351	6.345
ORB Extraction ( $\sigma$ )	61.54	57.737	58.214	58.98	58.148
Stereo Matching ( $\sigma$ )	0.54303	0.63088	0.68821	0.74869	0.6854


DATASET - 07

Radius multiplier	1.0	2.0	4.0	8.0	16.0
Translation error	0.48	0.44	0.47	0.46	0.47
Rotational error	0.24	0.23	0.25	0.26	0.26
ATE (m)	0.52	0.49	0.5	0.54	0.46
RPE (m)	0.014	0.013	0.014	0.014	0.014
RPE (deg)	0.04	0.039	0.041	0.040	0.040
ORB Extraction (ms)	6.305	6.3717	6.356	6.280	6.679
Stereo Matching (ms)	3.087	3.573	4.072	4.896	6.679
ORB Extraction ( $\sigma$ )	29.31	28.58	29.45	28.65	29.87
Stereo Matching ( $\sigma$ )	0.535	0.526	0.642	0.689	0.688

ATE minore




# Accelerazione algoritmo di Stereo Matching

- Motivazioni accelerazione dell'algoritmo
- Procedimento 
- Risultati



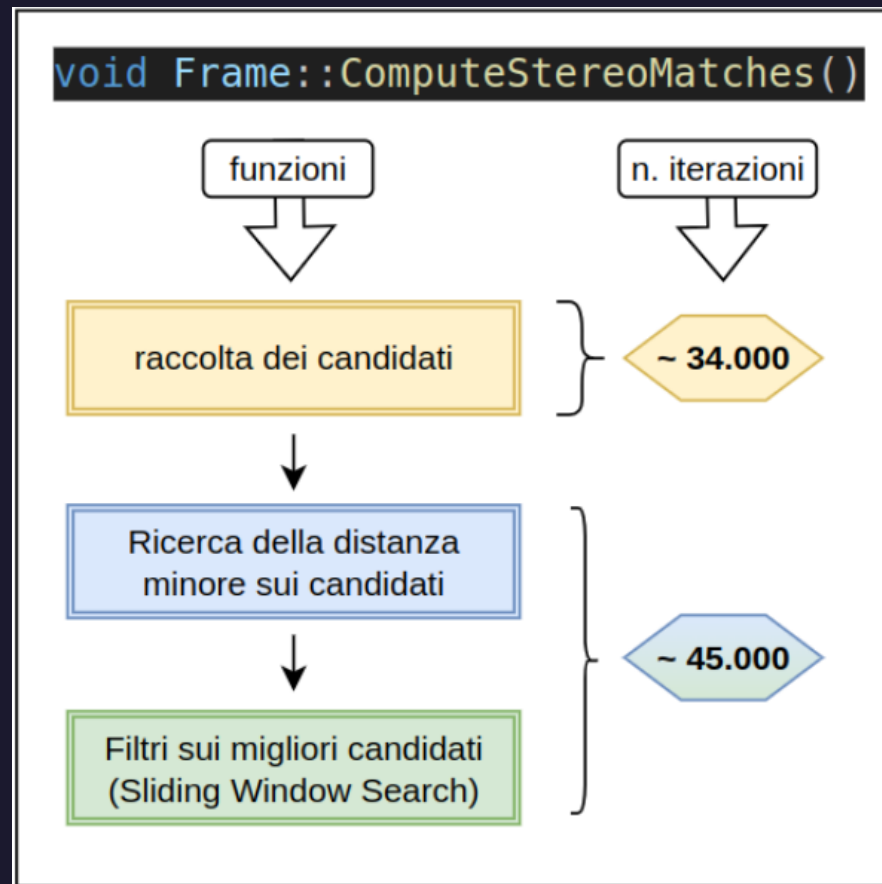
# Accelerazione algoritmo di Stereo Matching

- Motivazioni accelerazione dell'algoritmo
- Procedimento 
- Risultati

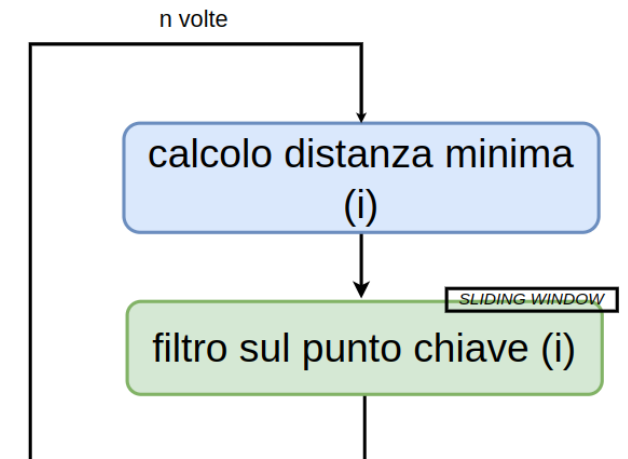
## Analisi funzione stereo matching ①

CICLO FOR {


CICLO FOR {



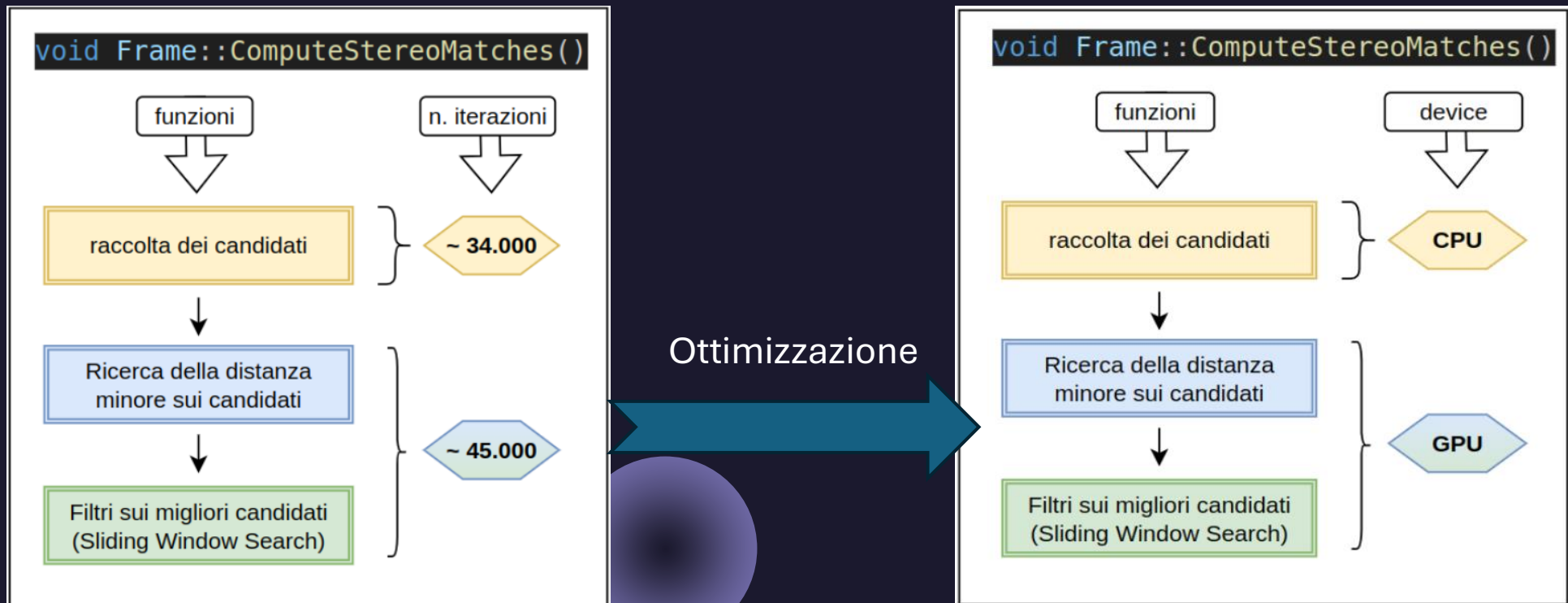
●  $n = \text{numero di punti chiave}$




# Accelerazione algoritmo di Stereo Matching

- Motivazioni accelerazione dell'algoritmo
- Procedimento 
- Risultati

Parallelizzazione della funzione  
usando CUDA 2



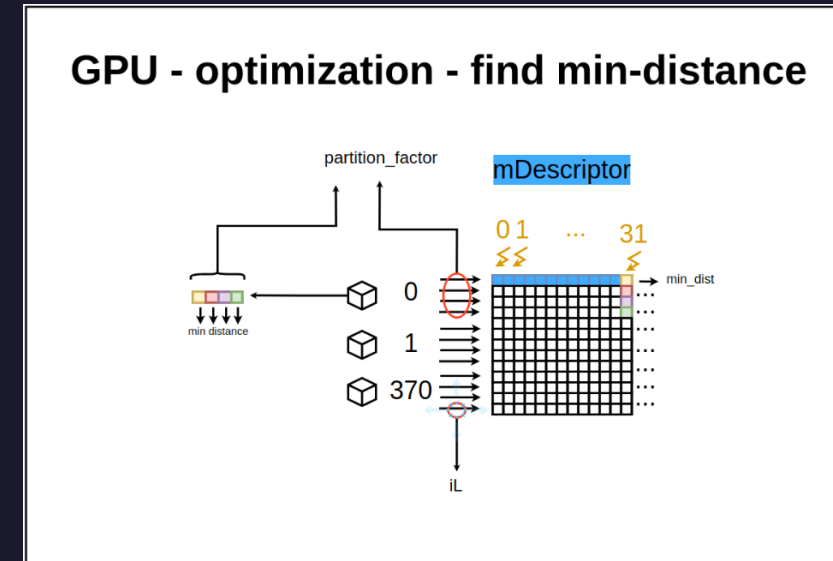
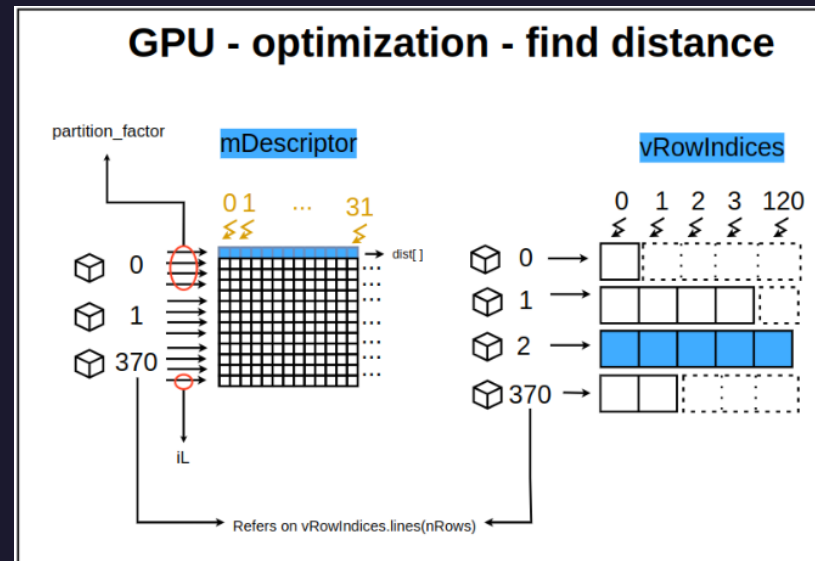
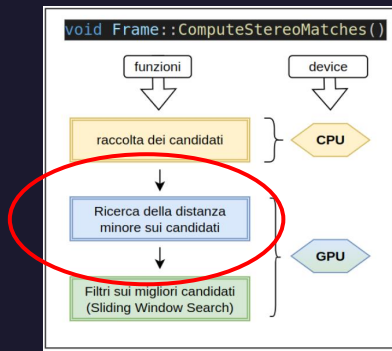
# Accelerazione algoritmo di Stereo Matching

- Motivazioni accelerazione dell'algoritmo
- Procedimento 
- Risultati

## Parallelizzazione della funzione usando CUDA ②


2

# KERNEL CUDA - 1





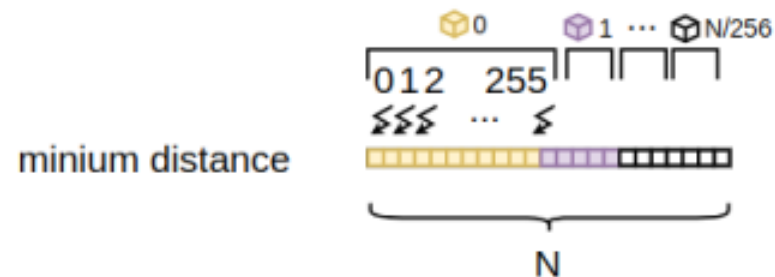
# Accelerazione algoritmo di Stereo Matching

- Motivazioni accelerazione dell'algoritmo
- Procedimento 
- Risultati

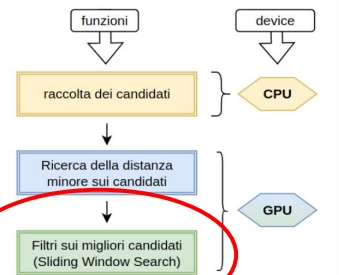
Parallelizzazione della funzione  
usando CUDA 2

KERNEL CUDA - 2

## GPU - optimization - sliding window



```
void Frame::ComputeStereoMatches()
```



# Accelerazione algoritmo di Stereo Matching

- Motivazioni accelerazione dell'algoritmo
- Procedimento
- Risultati

Problemi da gestire

2



cv::Mat

ORBmatcher::DescriptorDistance

## RISCRITTURA DI FUNZIONI SPECIFICHE

```
// Funzione che calcola la distanza tra 2 vettori
__device__ int DescriptorDistance(const unsigned char* a, const unsigned char* b){

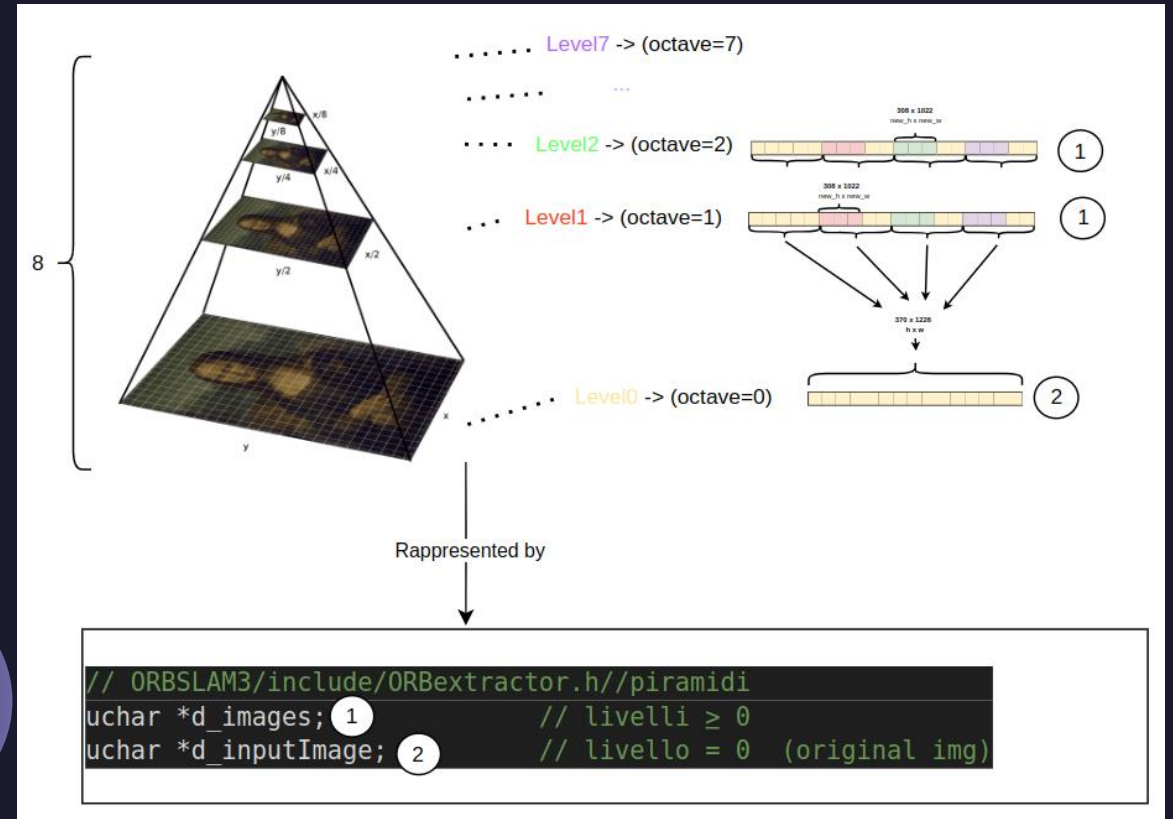
    int dist=0;

    const int32_t* a_int = reinterpret_cast<const int32_t*>(a);
    const int32_t* b_int = reinterpret_cast<const int32_t*>(b);


    for(int i=0; i<8; i++) {
        unsigned int v = a_int[i] ^ b_int[i];
        v = v - ((v >> 1) & 0x55555555);
        v = (v & 0x33333333) + ((v >> 2) & 0x33333333);
        dist += (((v + (v >> 4)) & 0xF0F0F0F) * 0x1010101) >> 24;
    }

    return dist;
}
```

## TRASFORMAZIONE DI STRUTTURE DATI

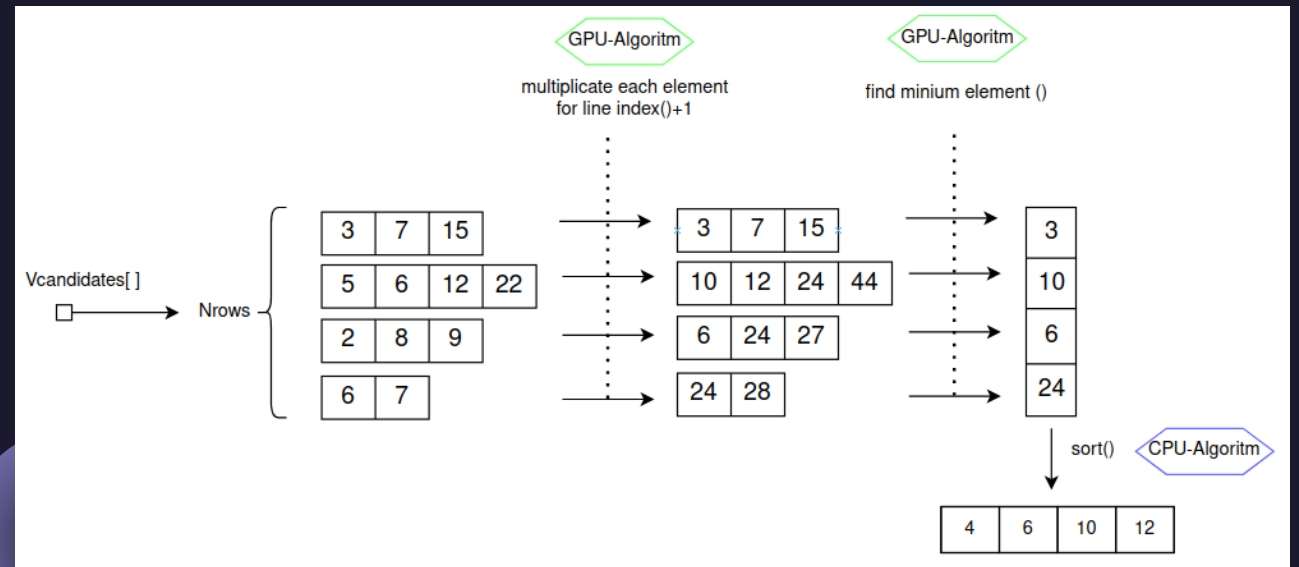
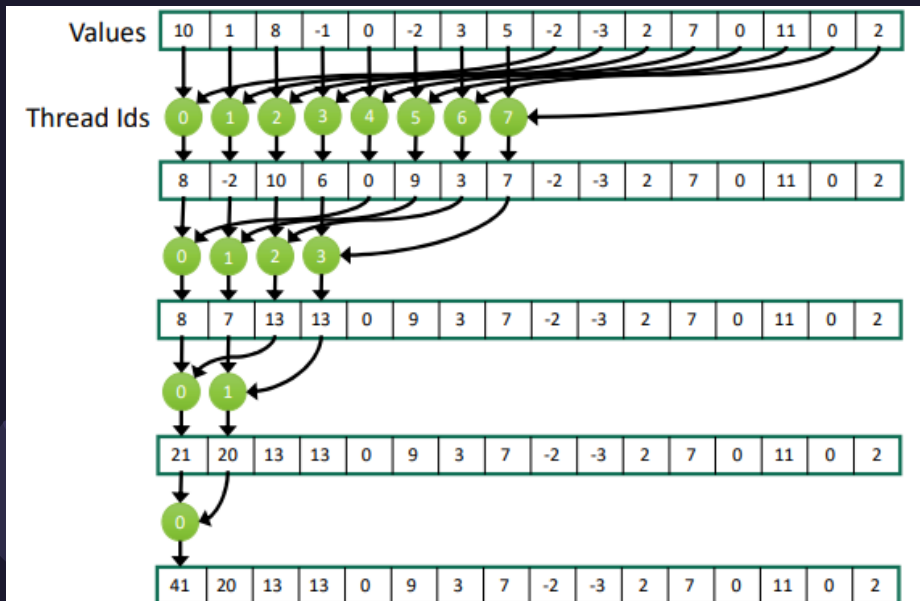


# Accelerazione algoritmo di Stereo Matching

- Motivazioni accelerazione dell'algoritmo
- Procedimento 
- Risultati

Parallelizzazione della funzione 2 usando CUDA

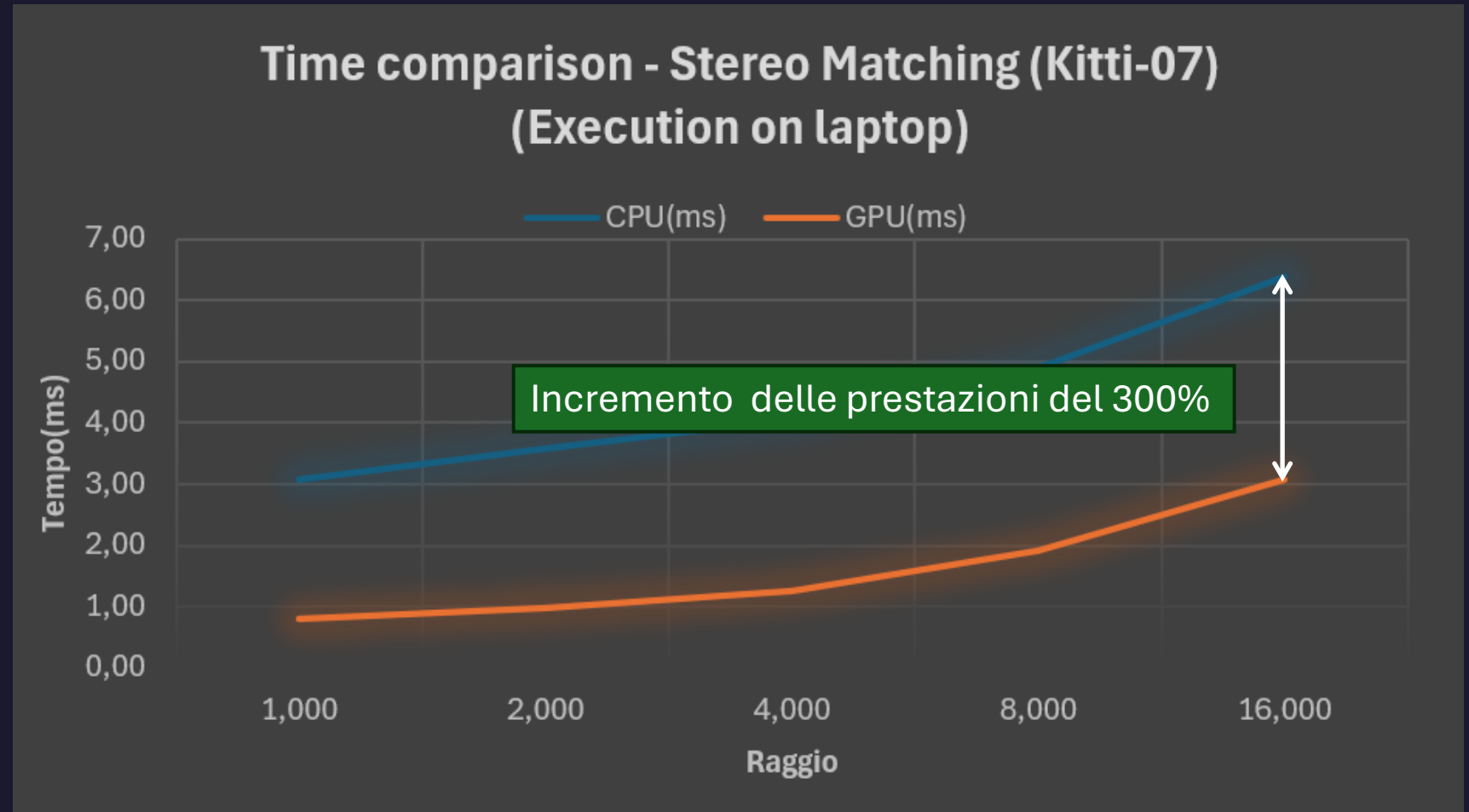
ADATTAMENTO DEL PROGRAMMA AL PARADIGMA PARALLELO




# Accelerazione algoritmo di Stereo Matching

- Motivazioni accelerazione dell'algoritmo
- Procedimento
- Risultati ←

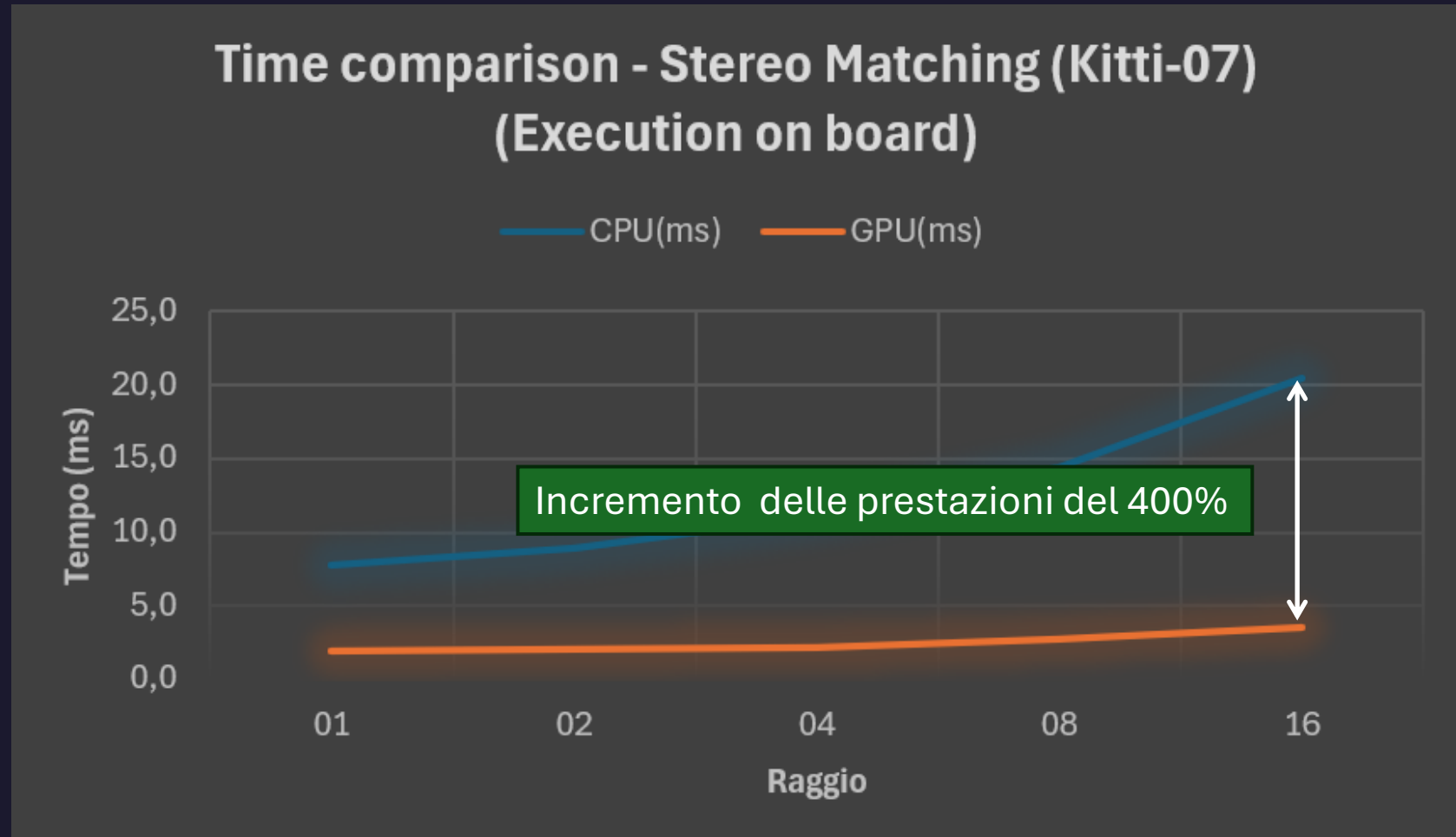
DEVICE - LAPTOP



# Accelerazione algoritmo di Stereo Matching




- Motivazioni accelerazione dell'algoritmo
- Procedimento
- Risultati 

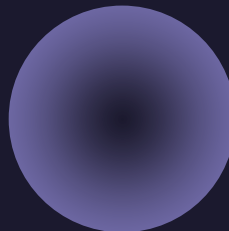
DEVICE - BOARD





# Conclusioni

- Aumento delle prestazioni temporali del sistema 
- Aumento delle precisione del sistema 
- Apertura verso possibili esperimenti sfruttando la diminuzione dei tempi di calcolo 



Grazie per  
l'attenzione

