

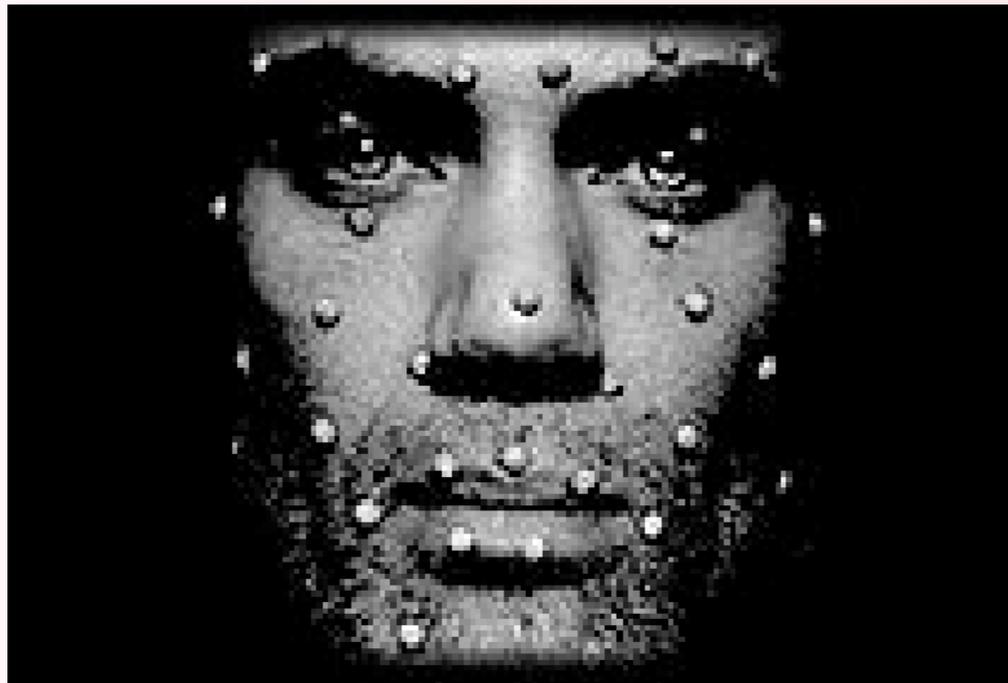


Motion capture distribuito

Alberto Tonello

Luca Fardin

Lorenzo Corso

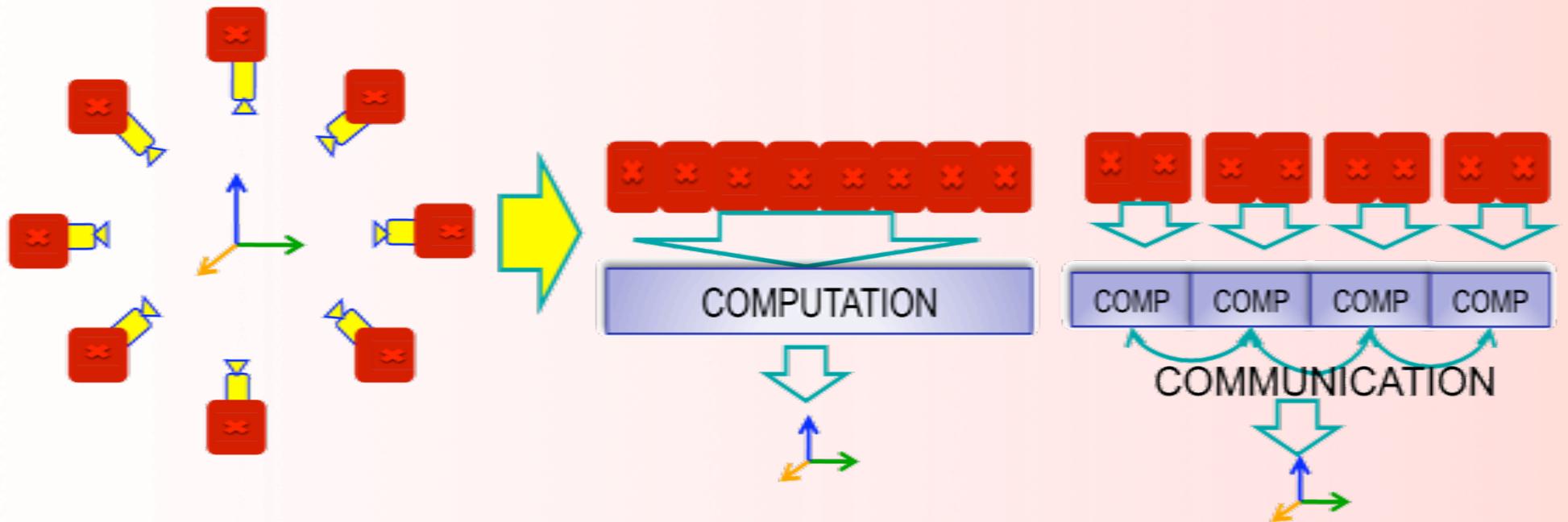


Obiettivo

- Implementare il passive motion capture in modo distribuito
- Rappresentare l'incertezza nell'errore di ricostruzione dei marker

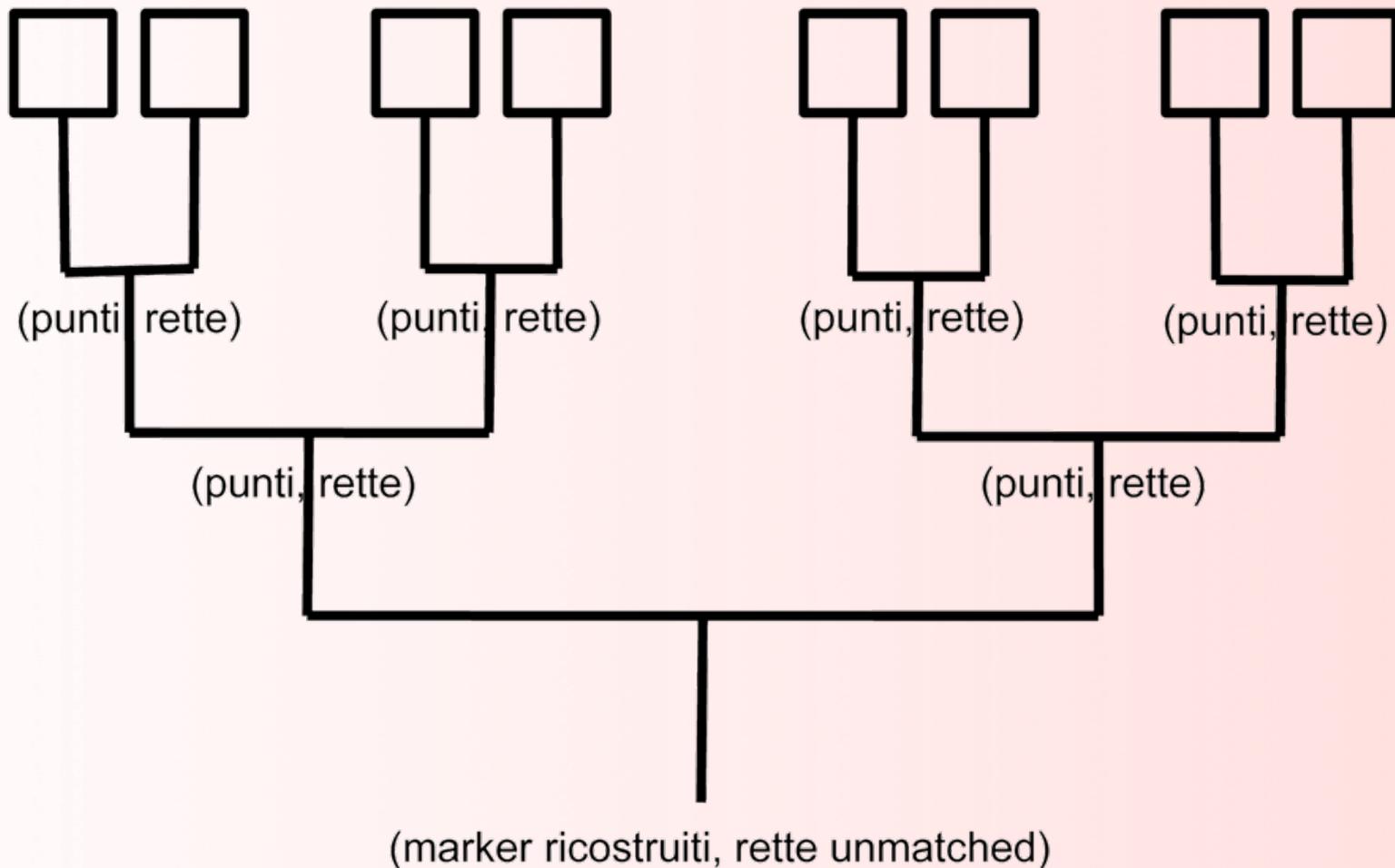
Approccio

Centralizzato o Distribuito



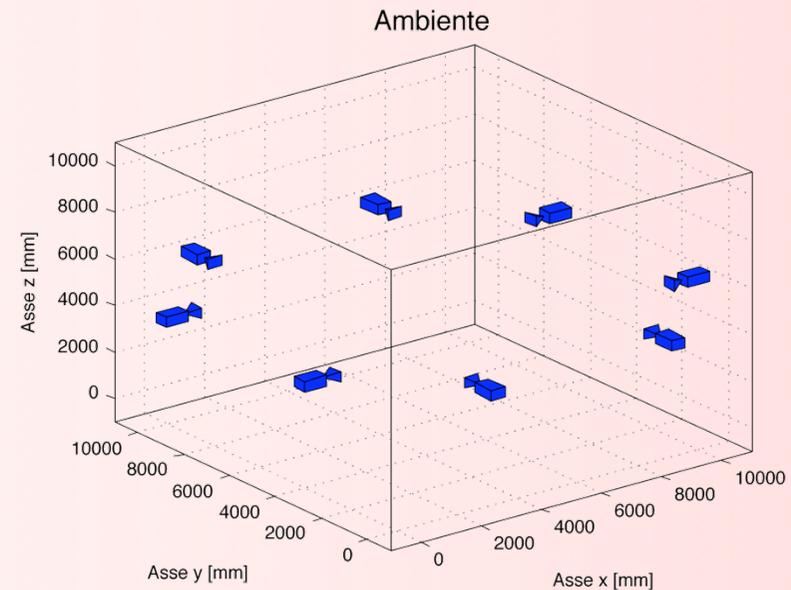
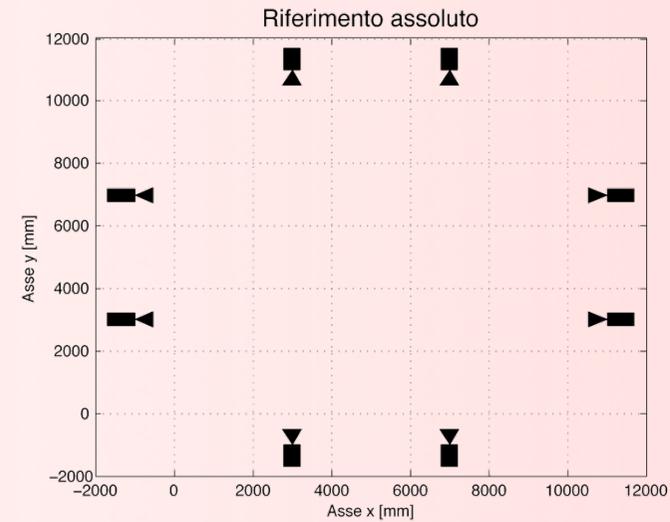
Approccio distribuito

Gerarchia di comunicazione ad albero



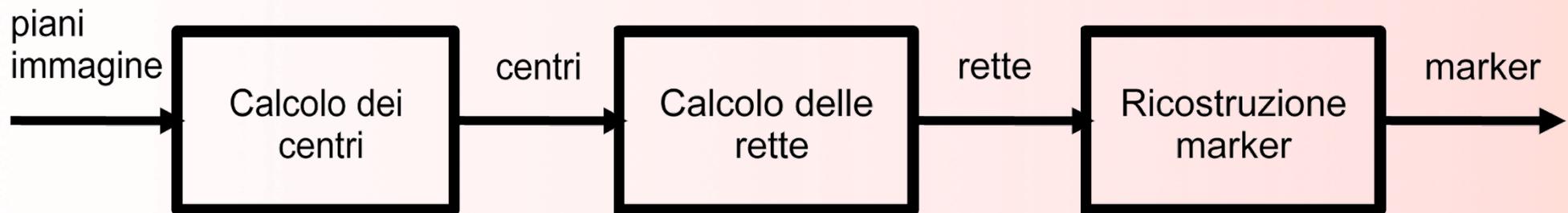
Approccio distribuito

- Analisi nel piano
- Analisi nello spazio



Progetto nel piano

- Calcolo dei centri
- Calcolo delle rette
- Ricostruzione dei marker nel piano
- Incertezza





Calcolo dei centri

- Ingressi:
 - Piano immagine
 - Dimensione pixel
 - Distanza tra serie di pixel
- Uscite:
 - Posizione dei centri sul piano immagine
 - Numero dei pixel di ogni centro

Algoritmo implementato

Primo passaggio



Secondo passaggio





Calcolo delle rette

- Matrice di trasformazione

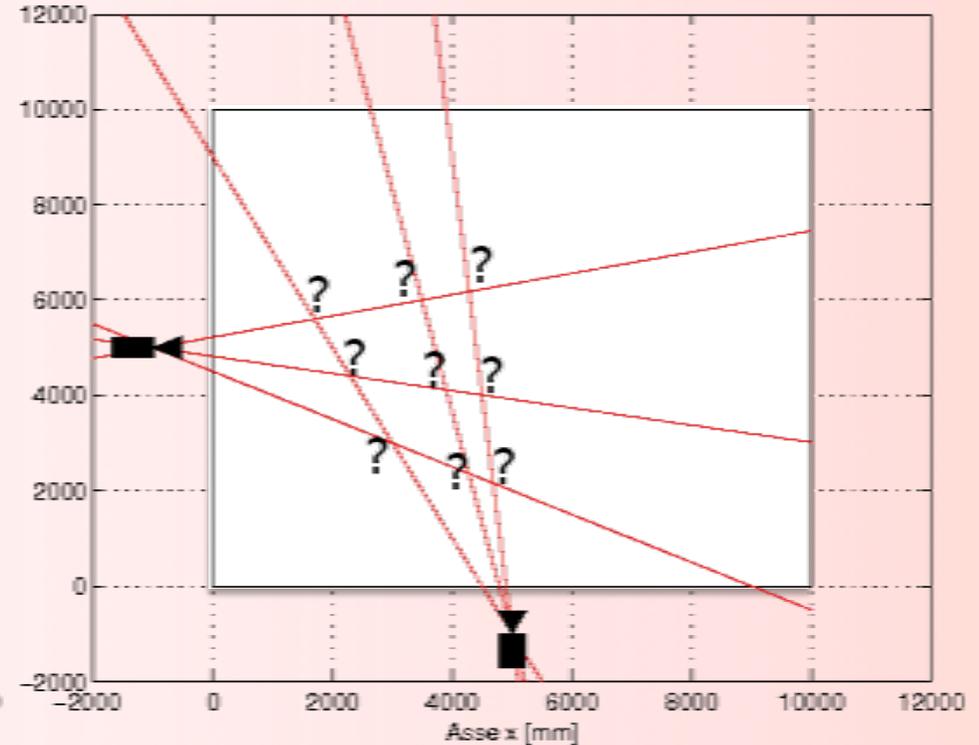
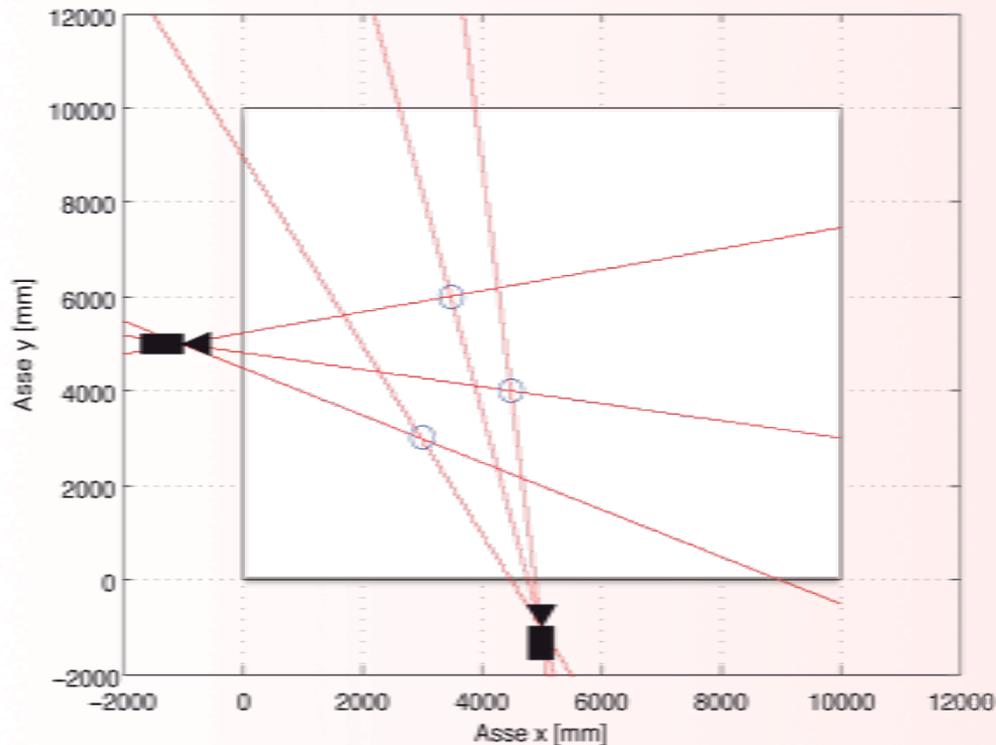
$$T_{21} = \begin{bmatrix} \cos(\alpha) & -\text{sen}(\alpha) & 0 & x_t \\ \text{sen}(\alpha) & \cos(\alpha) & 0 & y_t \\ 0 & 0 & 1 & z_t \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} x_{\text{assoluto}} \\ y_{\text{assoluto}} \\ z_{\text{assoluto}} \\ 1 \end{bmatrix} = T_{21} \cdot \begin{bmatrix} x_{\text{coordinate}} \\ y_{\text{coordinate}} \\ z_{\text{coordinate}} \\ 1 \end{bmatrix}, \quad \begin{bmatrix} x_{f,\text{assoluto}} \\ y_{f,\text{assoluto}} \\ z_{f,\text{assoluto}} \\ 1 \end{bmatrix} = T_{21} \cdot \begin{bmatrix} x_f \\ y_f \\ z_f \\ 1 \end{bmatrix}$$

- Equazione implicita della retta

$$ax + by + c = 0$$

Ricostruzione dei marker

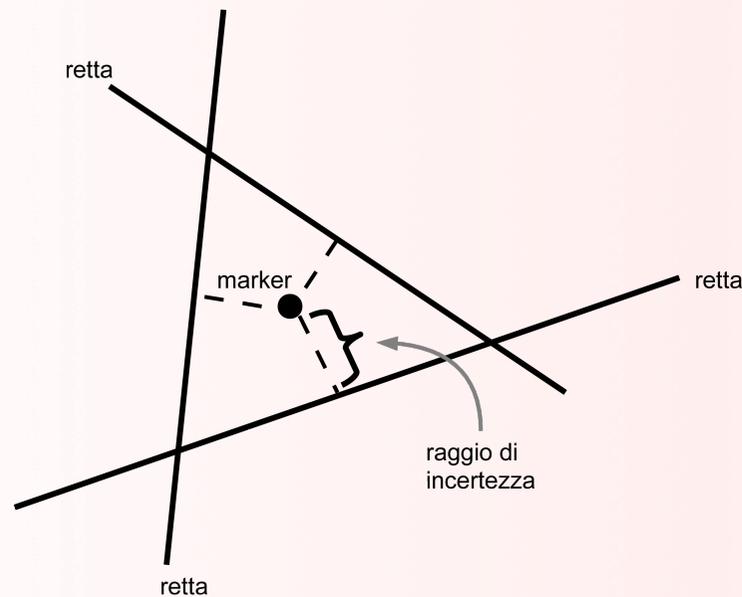


- Necessità di avere almeno tre raggi per la ricostruzione
- Algoritmo ai minimi quadrati per la ricostruzione

Minimi quadrati

- Individuazione punto di minima distanza

$$\min_{x,y} \sum d_i^2 \text{ con } d_i^2 = \frac{(a_i x + b_i y + c_i)^2}{a_i^2 + b_i^2}$$



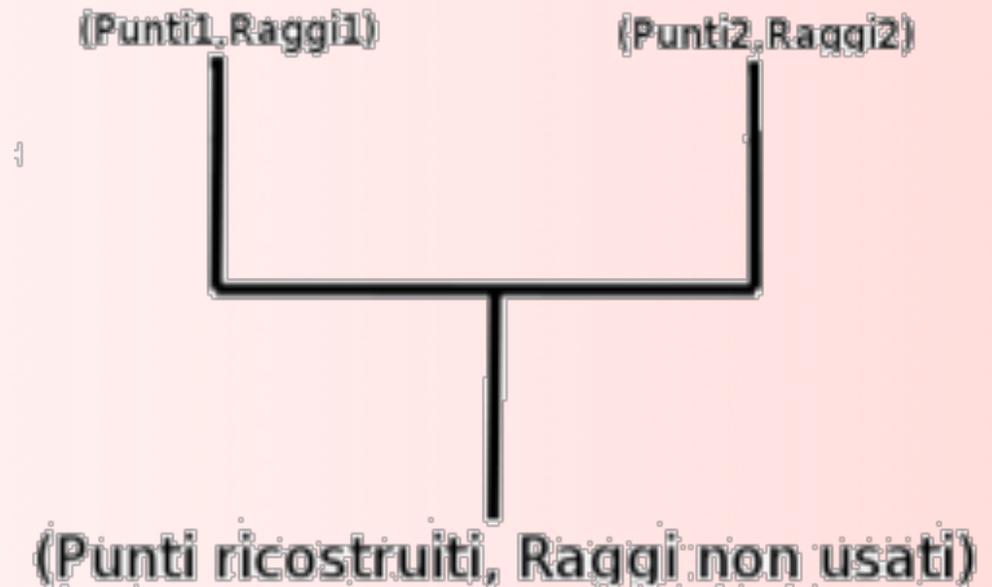


Minimi quadrati

- Esiste soluzione in forma chiusa
- Possibilità d'implementare un algoritmo generale che processa un numero arbitrario di rette

Struttura della funzione

- Ingressi:
 - Raggi
 - Punti
 - Epsilon
 - Specifiche ambiente
- Uscite:
 - Punti
 - Raggi non utilizzati





Struttura della funzione

- Controllo che raggio non corrisponda ad un marker già ricostruito, in tal caso due approcci:
 - i. Eliminazione raggio
 - ii. Ricalcolo punto con tutti i raggi
- Ricostruzione mediante minimi quadrati
- Successivamente possibile media di eventuali marker ricostruiti doppi



Minimi quadrati pesati

- Riduzione errore ricostruzione pesando diversamente i residui

$$\min_{x,y} \sum_i \alpha_i d_i^2 \text{ con } \sum_i \alpha_i = 1$$

Stima dei pesi valutando i piani immagine

$$\alpha_i = \frac{np_i}{\sum_j np_j}$$

- Peso maggiore ai raggi generati da marker più vicini alla telecamera



Funzione centralizzata

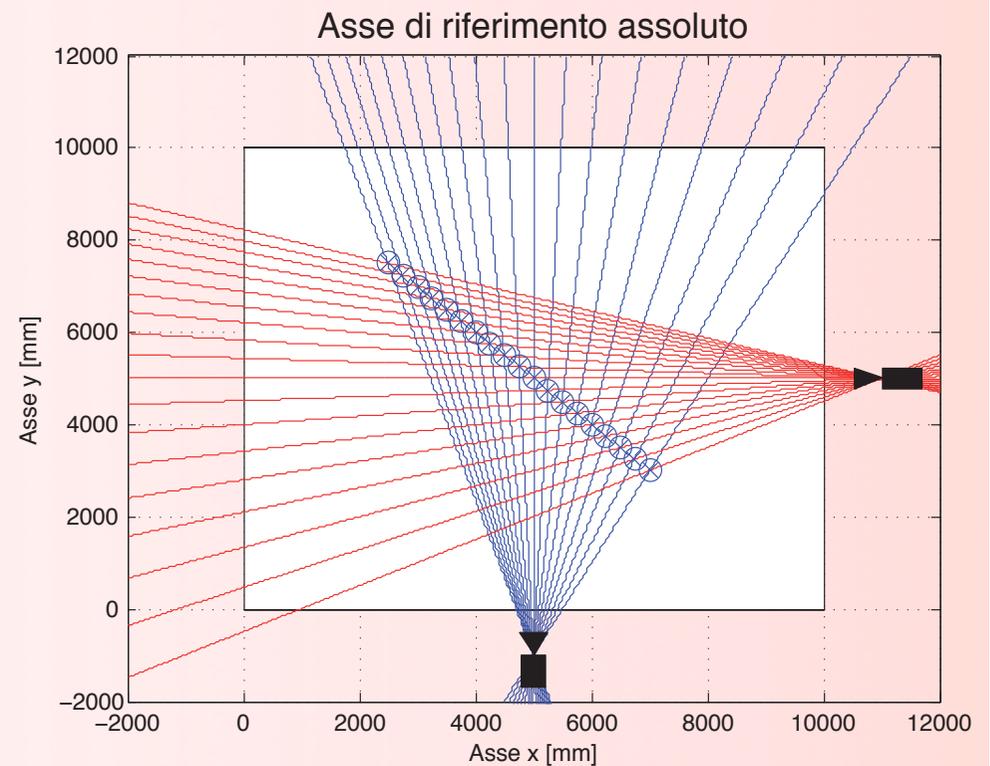
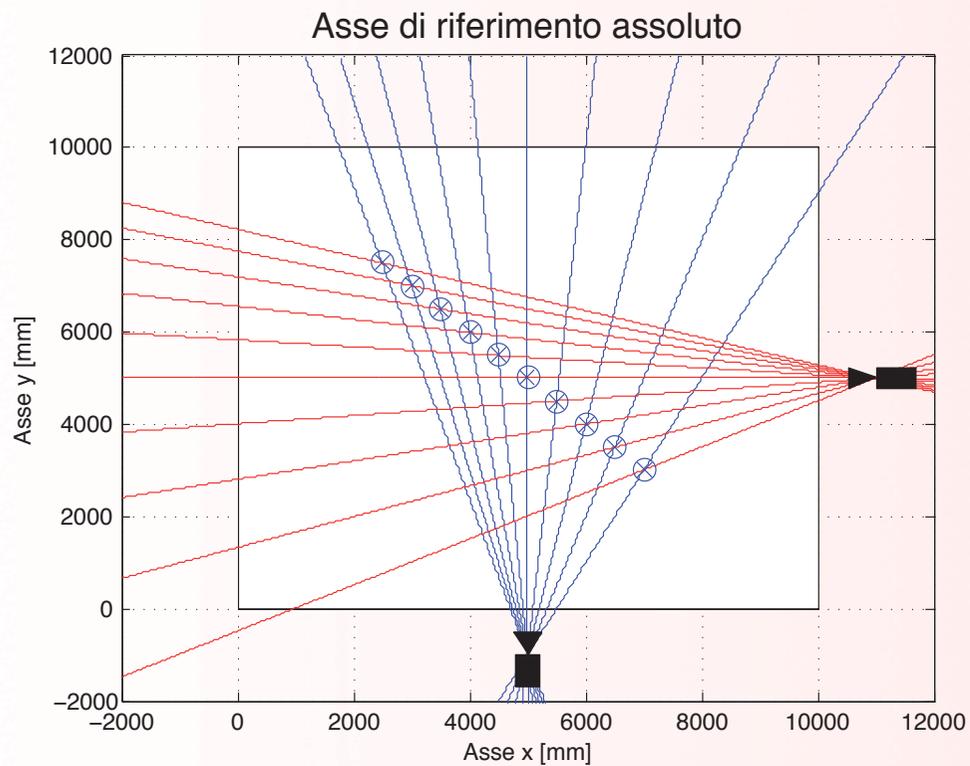
- I dati sono processati in modo centralizzato
- Necessità di utilizzare almeno quattro raggi
- Nessuna eliminazione dei raggi

Conclusioni

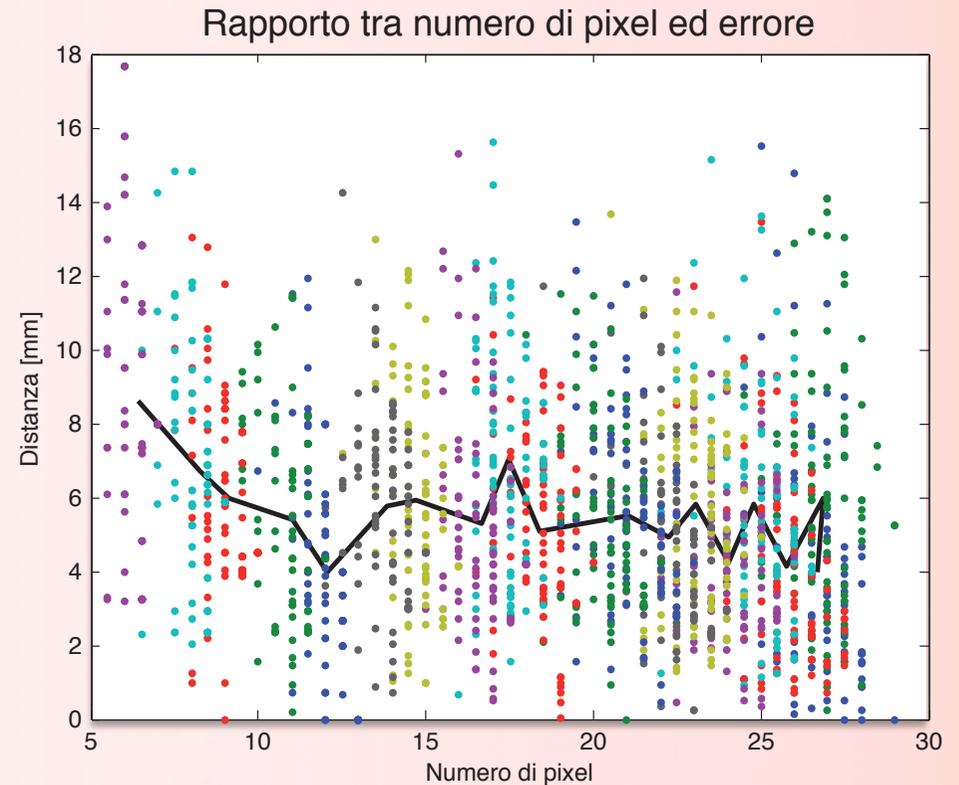
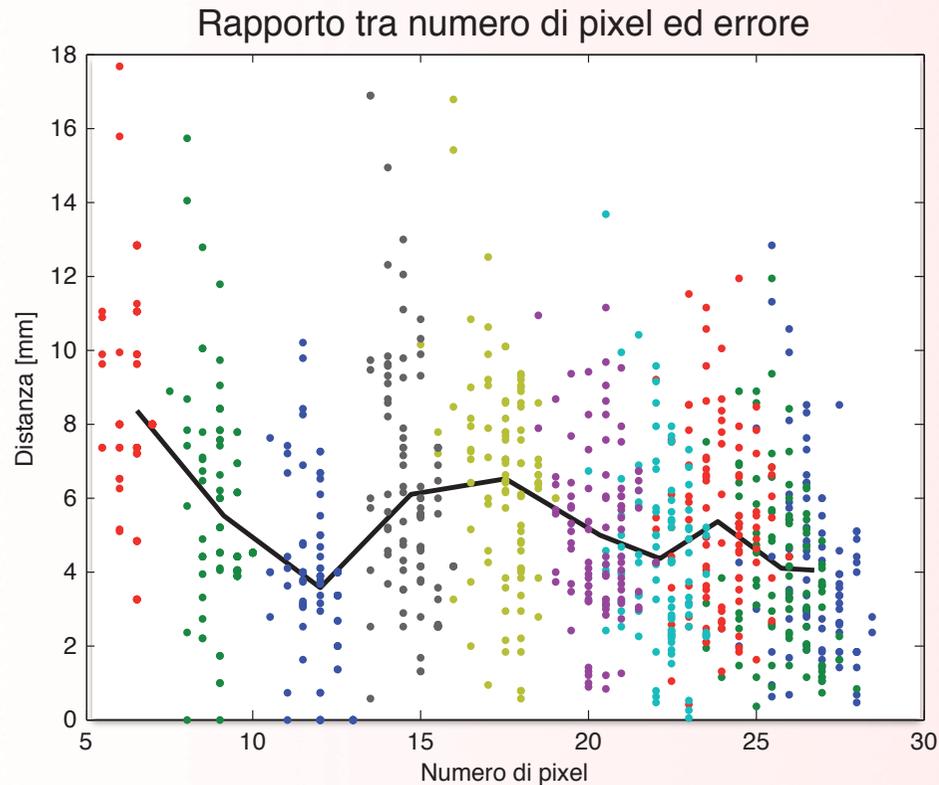
Tipo Ricostruzione	Errore medio	Tempo medio esecuzione
Minimi Quadrati, Elimina Raggi	2.6779 [mm]	0.2430 [s]
Minimi Quadrati, Mantieni Raggi	1.9265 [mm]	0.2480 [s]
Minimi Quadrati Pesati Elimina Raggi	2.6687 [mm]	0.2532 [s]
Minimi Quadrati Pesati, Mantieni Raggi	1.9187 [mm]	0.2587 [s]
Approccio Centralizzato	4.5350 [mm]	43 [s]

Rapporto tra numero di pixel ed errore di ricostruzione

Simulazioni con dieci e con diciannove marker



Impossibilità di stabilire una relazione



- Piani immagine non adeguati
- Errore di ricostruzione legato alla posizione



Progetto nello spazio

- Calcolo dei centri
- Calcolo delle rette
- Ricostruzione dei marker nello spazio
- Incertezza





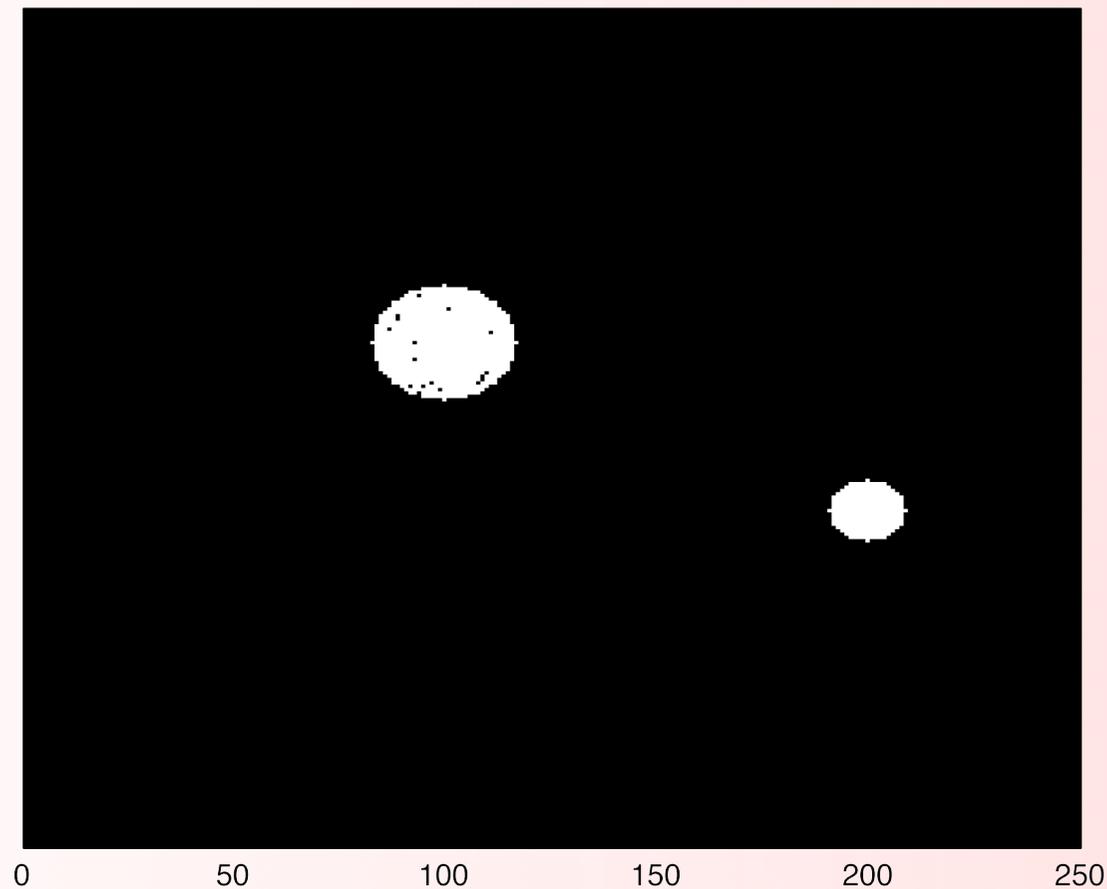
Calcolo dei centri

- Ingressi:
 - Piano immagine
 - Dimensione pixel
- Uscite:
 - Posizione dei centri sul piano immagine
 - Numero dei pixel di ogni centro

Algoritmo implementato

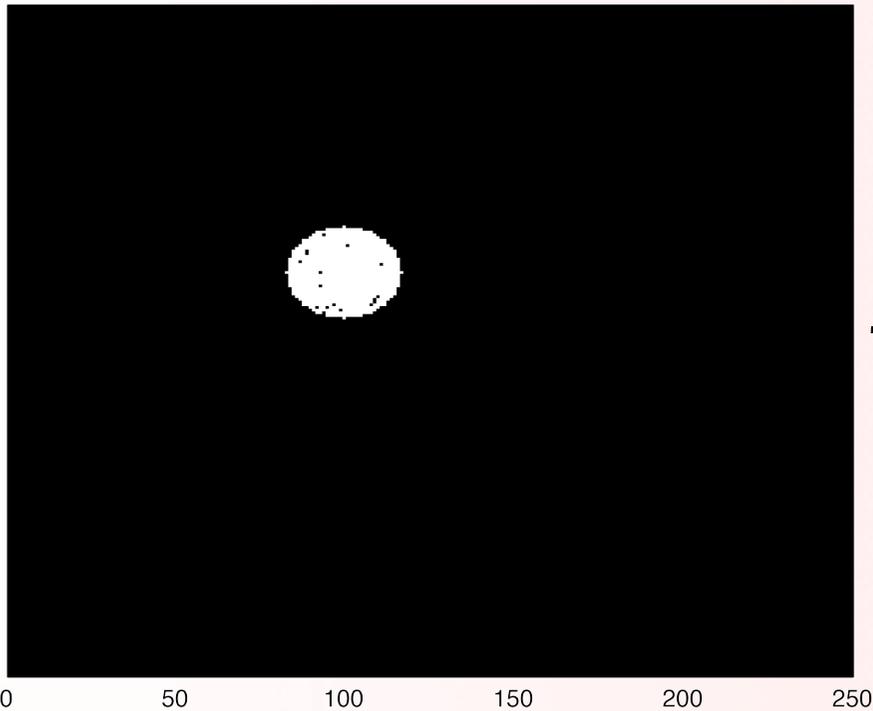
L'algoritmo è composto da due passaggi

Piano immagine

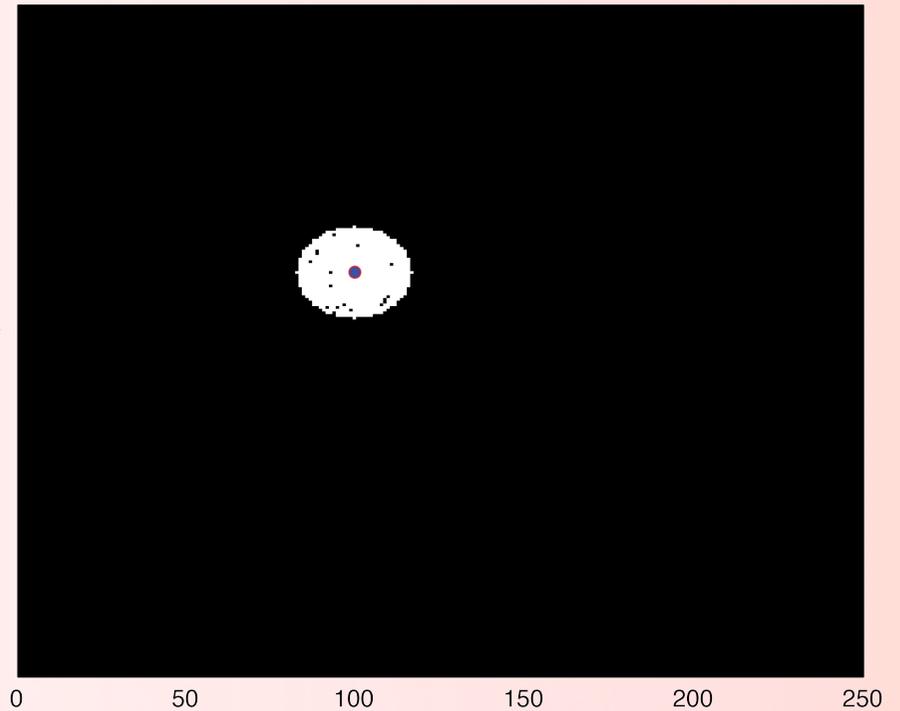


Per la prima proiezione

Primo passaggio

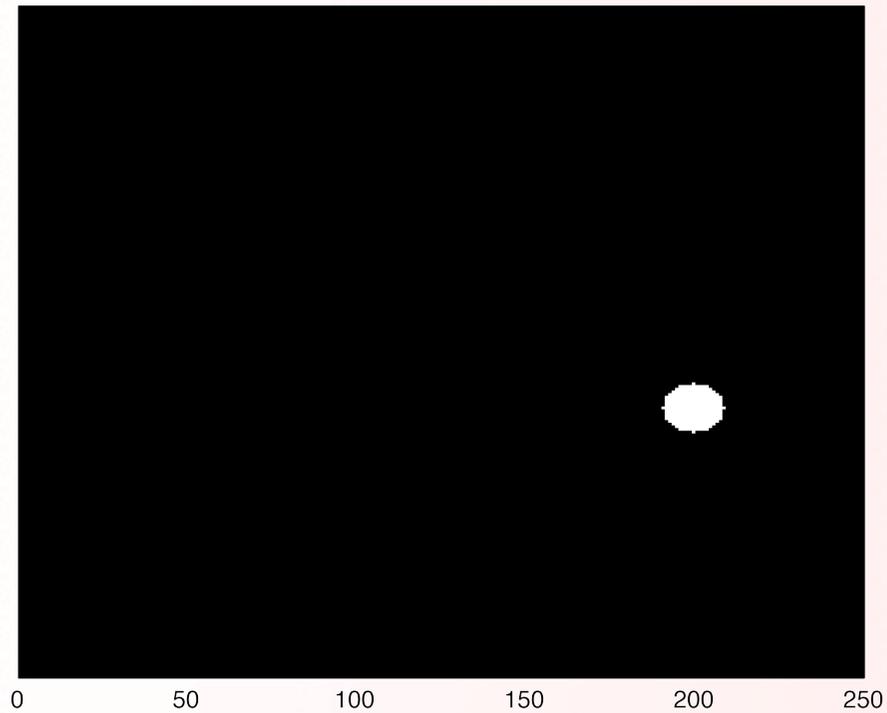


Secondo passaggio

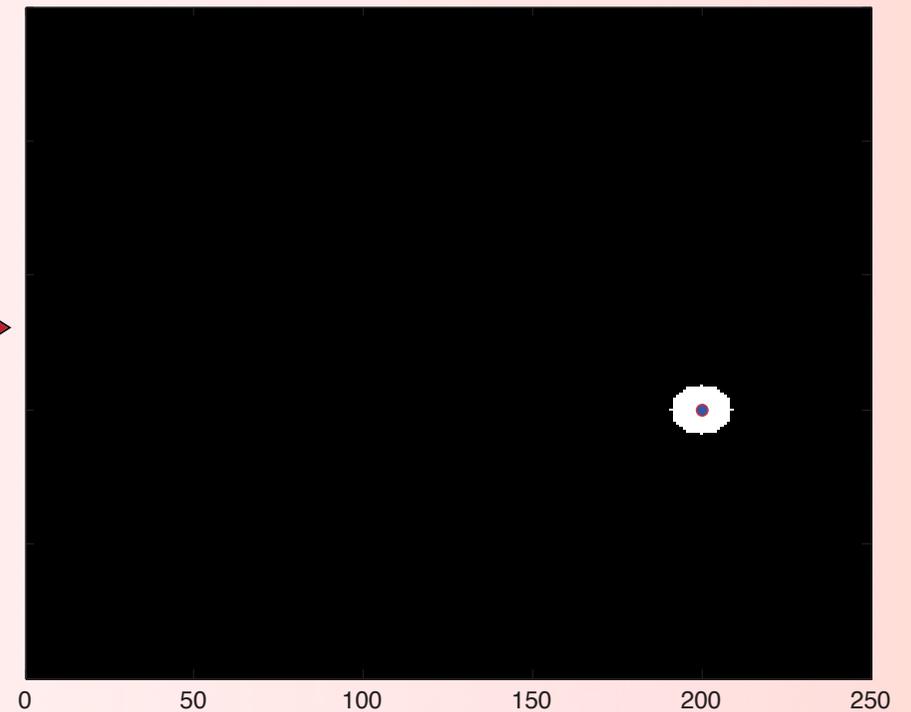


Per la seconda proiezione

Primo passaggio



Secondo passaggio





Calcolo delle rette

- **Ingressi:**
 - Posizione delle telecamere
 - Matrice di rotazione utilizzando la notazione di Cardano con rotazioni successive secondi gli assi $X \rightarrow Y \rightarrow Z$
 - Coordinate centri calcolate sul piano immagine
 - Altezza e larghezza piano immagine
 - Lunghezza focale telecamera
- **Uscite:**
 - Coordinate dei centri riportate nel sistema di riferimento assoluto.



Calcolo delle rette

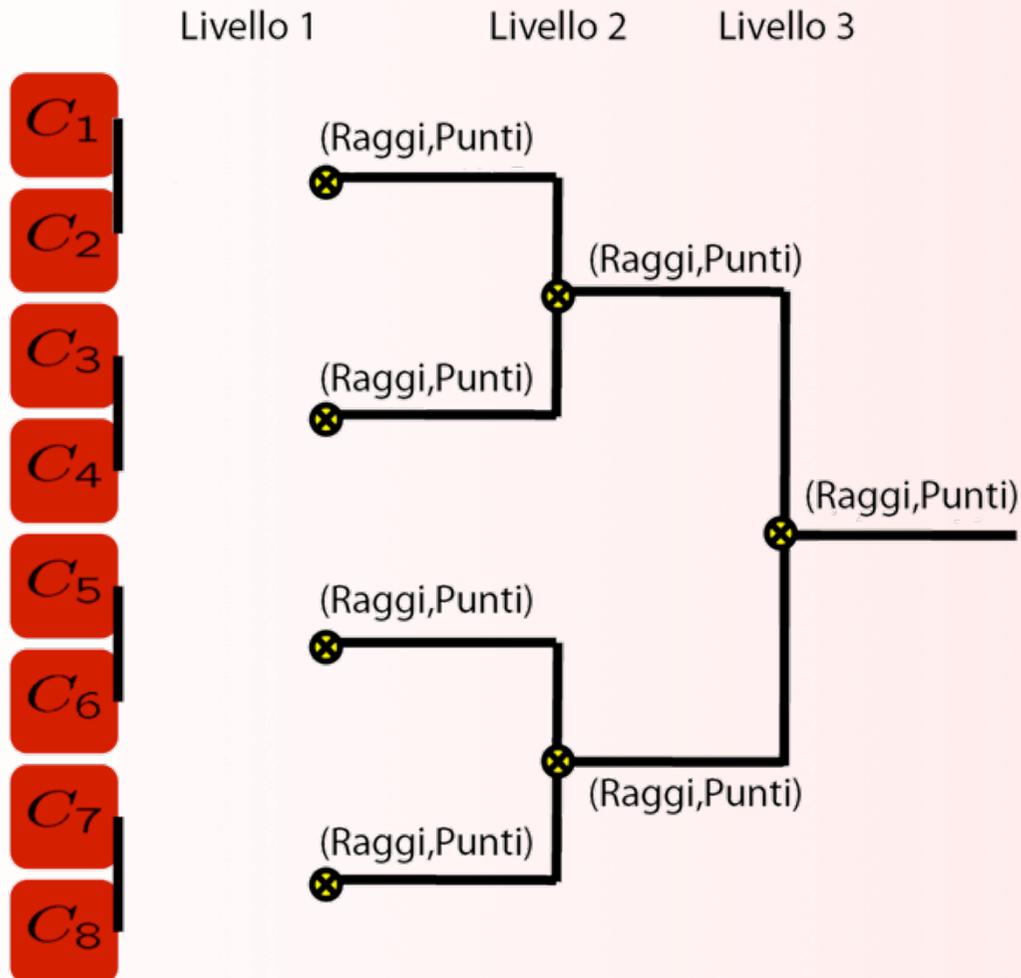
$$R_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha) & -\text{sen}(\alpha) \\ 0 & \text{sen}(\alpha) & \cos(\alpha) \end{bmatrix}, R_y = \begin{bmatrix} \cos(\beta) & 0 & \text{sen}(\beta) \\ 0 & 1 & 0 \\ -\text{sen}(\beta) & 0 & \cos(\beta) \end{bmatrix},$$

$$R_z = \begin{bmatrix} \cos(\delta) & -\text{sen}(\delta) & 0 \\ \text{sen}(\delta) & \cos(\delta) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$R = R_z \cdot R_y \cdot R_x$$

$$T = \begin{bmatrix} & R & & \\ & & x_t & \\ & & y_t & \\ & & z_t & \\ 0 & 0 & 0 & 1 \end{bmatrix} \xrightarrow{\text{red arrow}} \begin{bmatrix} x_{\text{assoluto}} \\ y_{\text{assoluto}} \\ z_{\text{assoluto}} \\ 1 \end{bmatrix} = T \cdot \begin{bmatrix} x_{\text{centri}} \\ y_{\text{centri}} \\ z_{\text{centri}} \\ 1 \end{bmatrix}$$

Ricostruzione marker



- Elimina rette
- Mediazione marker
- Ricostruzione pesata



Ricostruzione marker

- **Ingresso:**

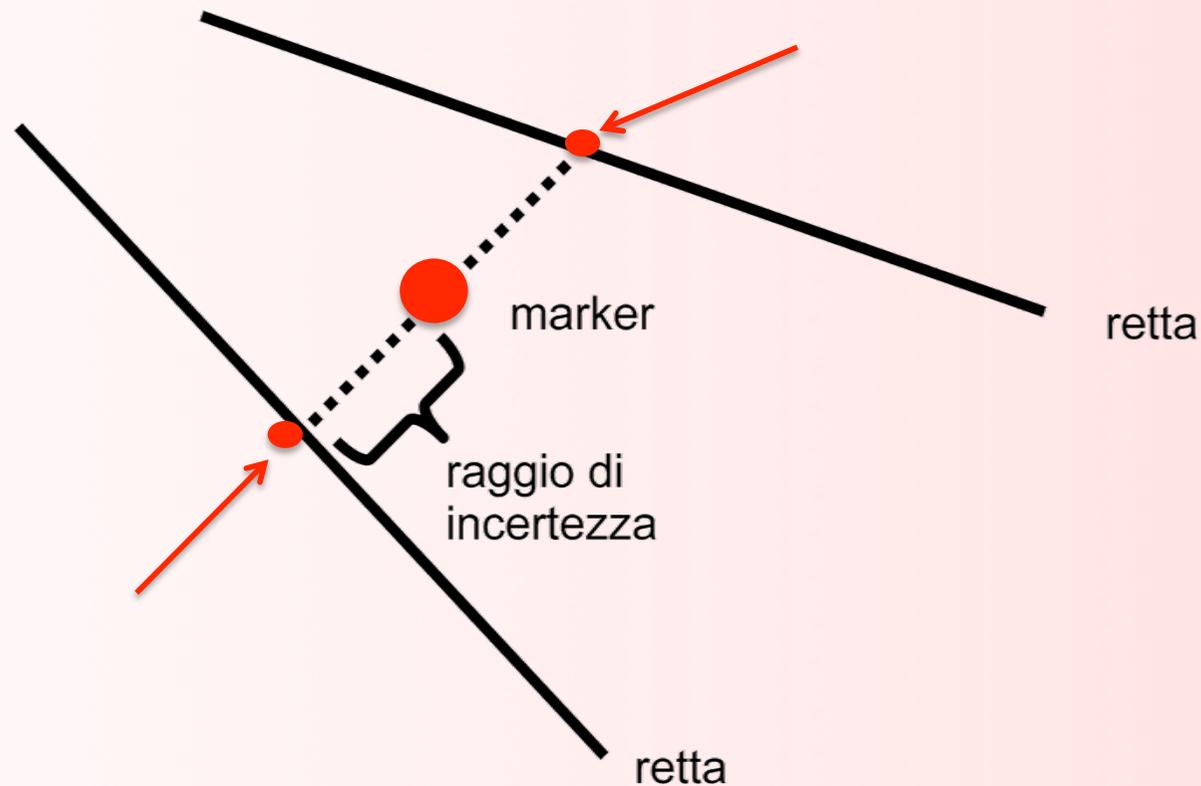
- Coordinate centri nel sistema di riferimento assoluto
- Posizione delle telecamere
- Distanza minima per la ricostruzione
- Distanza minima per la mediazione
- Numero di pixel dei raggi

- **Uscita:**

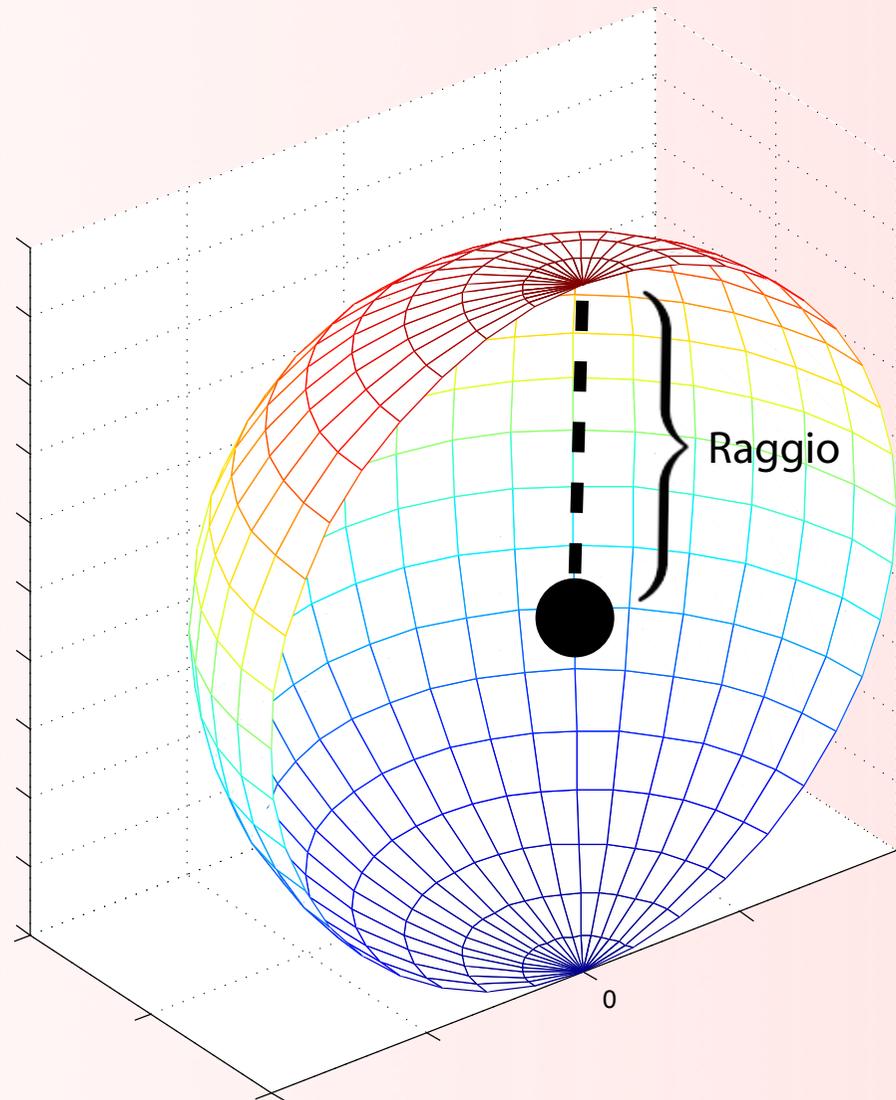
- Tempo di ricostruzione nei vari livelli albero
- Marker ricostruiti e loro distanza minima dalle rette che li hanno generati
- Rette non utilizzate per la ricostruzione

Ricostruzione marker

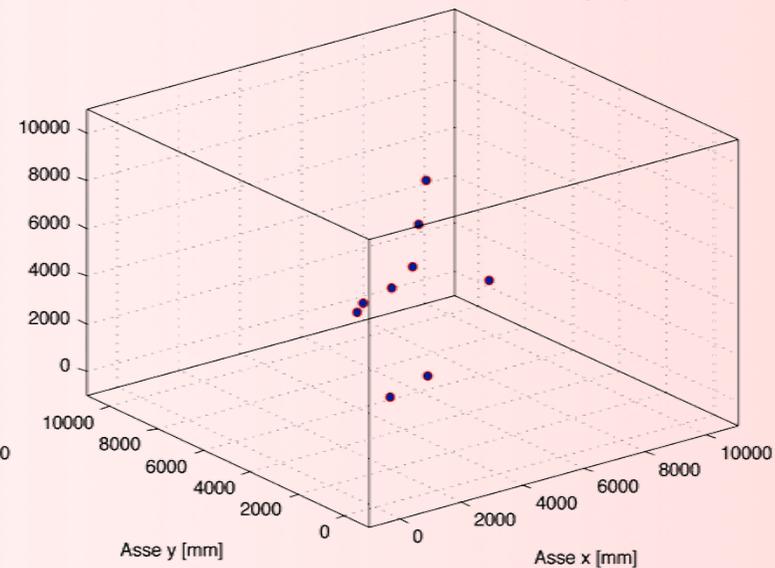
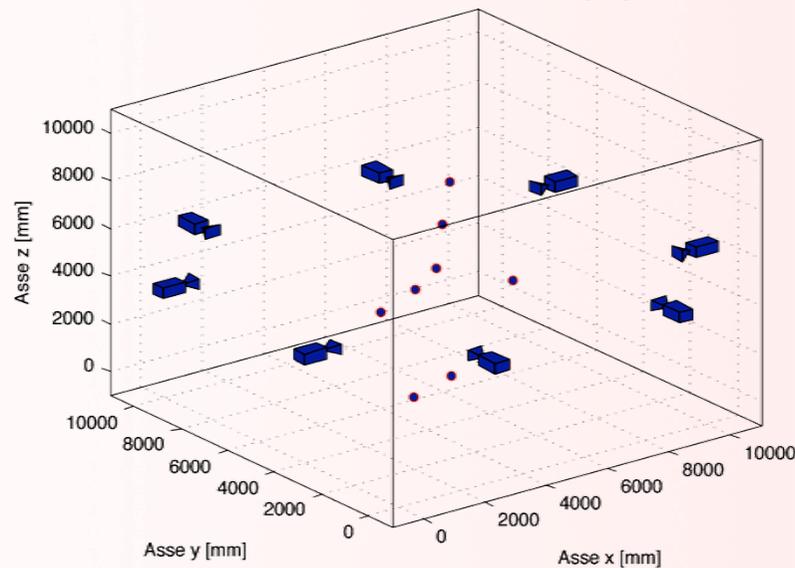
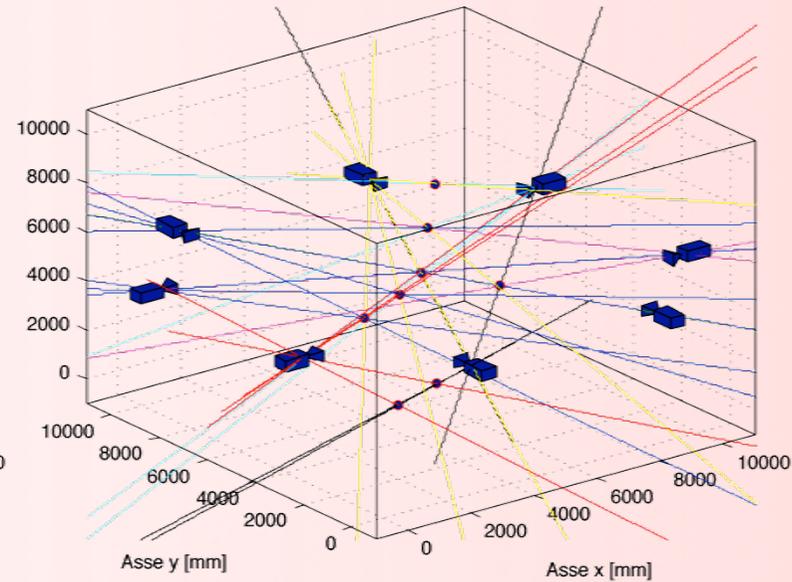
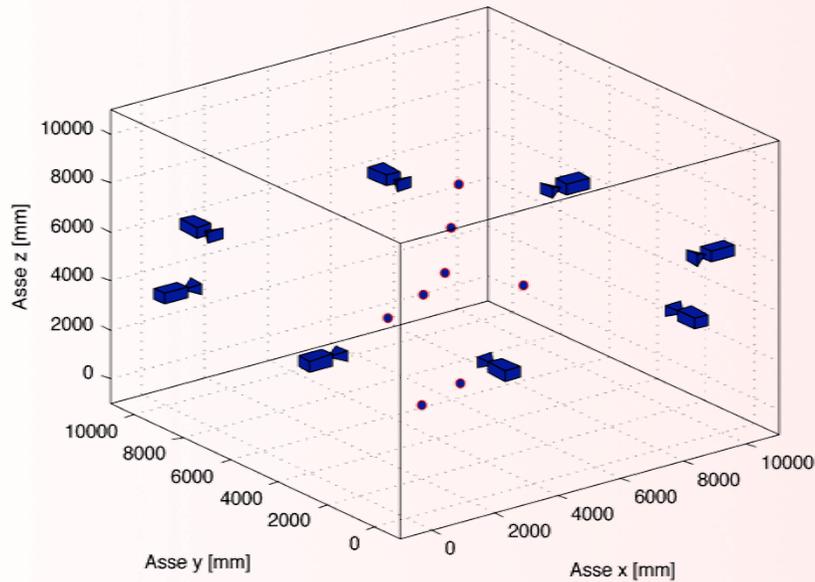
$$s : \begin{cases} x_1 = x_{0,1} + lt_1 \\ y_1 = y_{0,1} + mt_1 \\ z_1 = z_{0,1} + nt_1 \end{cases}, r : \begin{cases} x_2 = x_{0,2} + kt_2 \\ y_2 = y_{0,2} + vt_2 \\ z_2 = z_{0,2} + wt_2 \end{cases}$$



Volume d'incertezza



Algoritmo distribuito



Algoritmo distribuito

Algoritmo elimina rette

x	y	z	x_{ric}	y_{ric}	z_{ric}	errore	residuo
2000	2000	2000	1998,9	1999,4	2009,7	9,8179	0,4
5000	5000	5000	5000,1	5000,1	4999,3	6	1,3
6000	6000	6000	6004,1	5996,8	5999,2	5,4154	0,2
7000	7000	7000	7003,6	6997,9	6999,8	4,1814	3,2
4000	6000	3000	4000,1	6000,1	2999,8	4,7224	0,4
6000	3000	5000	5997,8	3004,7	4999,6	5,1862	0,9
4800	5600	3900	4802	5603	3903,4	4,7079	2
3900	2900	1800	3899,5	2901,9	1800,3	1,9934	0,5

Primo livello albero	Secondo livello albero	Terzo livello albero
0,002247	—	—
0,000385	0,0026667	0,003016
0,000589	0,000316	—
0,000892	—	—

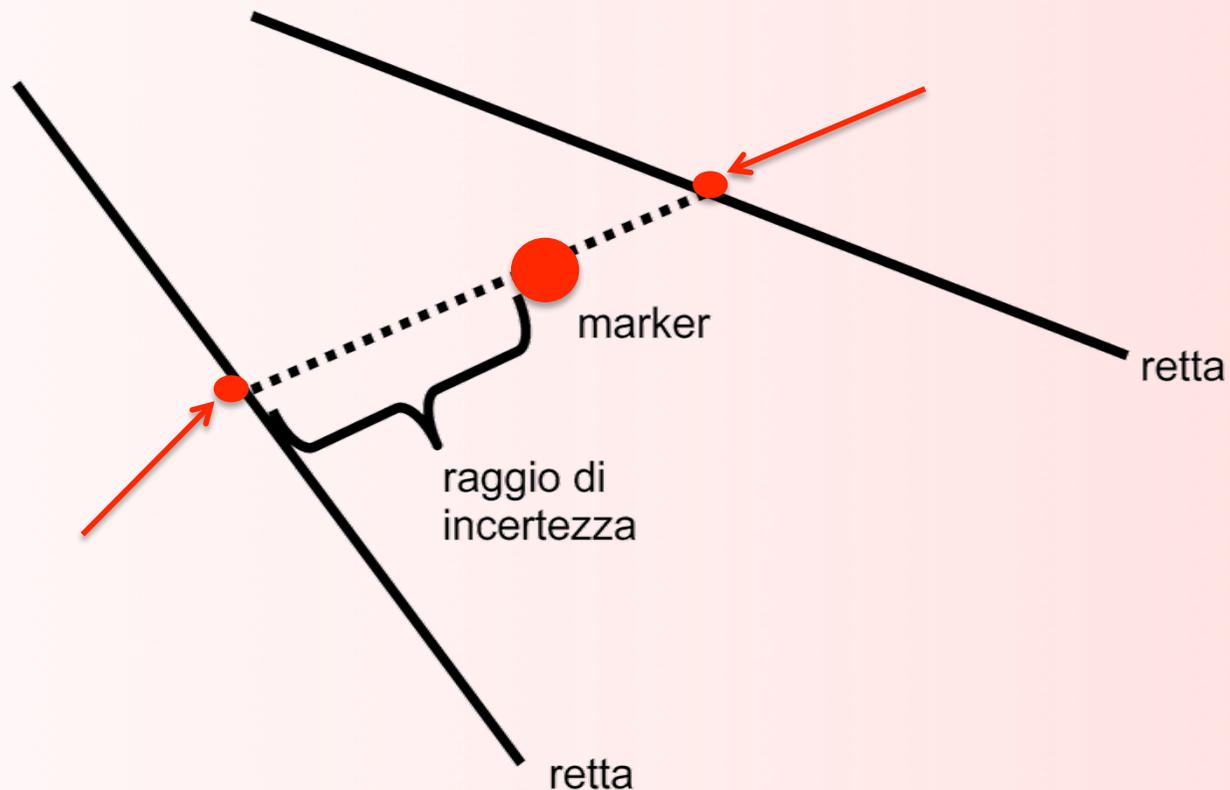
Algoritmo mediazione punti

x	y	z	x_{ric}	y_{ric}	z_{ric}	errore	residuo
2000	2000	2000	1998,9	1999,4	2009,7	9,8179	0,4
5000	5000	5000	5002,5	4999,5	4999,8	2,5634	11
6000	6000	6000	6004,7	5998,6	5999,5	4,9275	2
7000	7000	7000	7003,6	6997,9	6999,8	4,1814	3,2
4000	6000	3000	4002,4	5999,9	2999,7	2,4051	3,1
6000	3000	5000	5997,8	3004,7	4999,6	5,1862	0,9
4800	5600	3900	4802,5	5601,4	3903,5	4,5335	2,1
3900	2900	1800	3899,5	2901,9	1800,3	1,9934	0,5

Primo livello albero	Secondo livello albero	Terzo livello albero
0,021172	—	—
0,00334	0,024777	0,008887
0,000612	0,002178	—
0,00081	—	—

Algoritmo distribuito pesato

$$Punto = P \cdot \frac{num_pixel_s}{num_pixel_{tot}} + Q \cdot \frac{num_pixel_r}{num_pixel_{tot}}$$

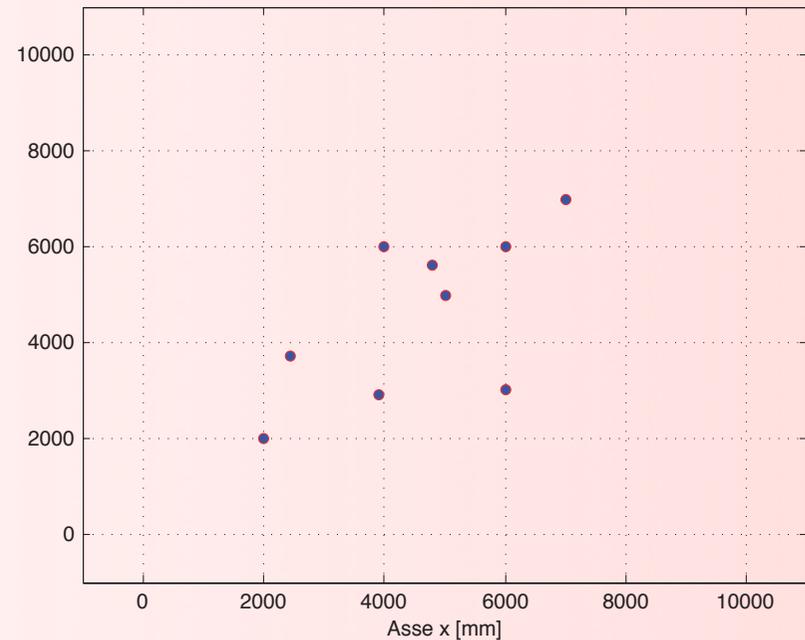
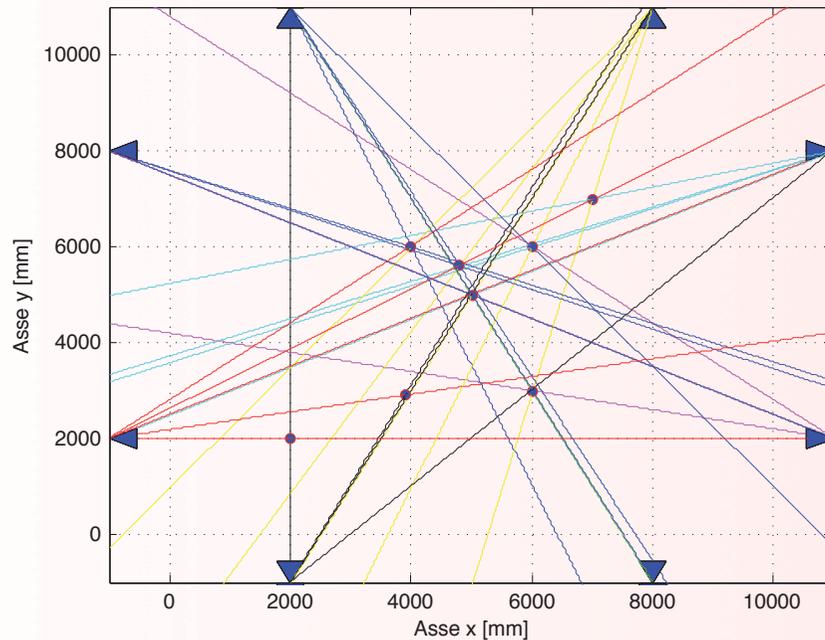


Algoritmo distribuito pesato

x	y	z	x_{ric}	y_{ric}	z_{ric}	errore	residuo
2000	2000	2000	1998,9	1999,4	2009,7	9,8179	0,2
5000	5000	5000	5002,5	4999,5	4999,7	2,5776	11,1
6000	6000	6000	6004,7	5998,6	5999,5	4,9275	2
7000	7000	7000	7003,6	6997,9	6999,8	4,1868	1,7
4000	6000	3000	4002,4	5999,9	2999,6	2,3962	3
6000	3000	5000	5997,8	3004,7	4999,3	5,2176	0,7
4800	5600	3900	4802,5	5601,4	3903,5	4,5238	1,9
3900	2900	1800	3899,6	2902	1800,3	2,0291	0,7

Primo livello albero	Secondo livello albero	Terzo livello albero
0,020784	—	—
0,015404	0,028904	0,004091
0,000877	0,003912	—
0,001261	—	—

Visione dall'alto



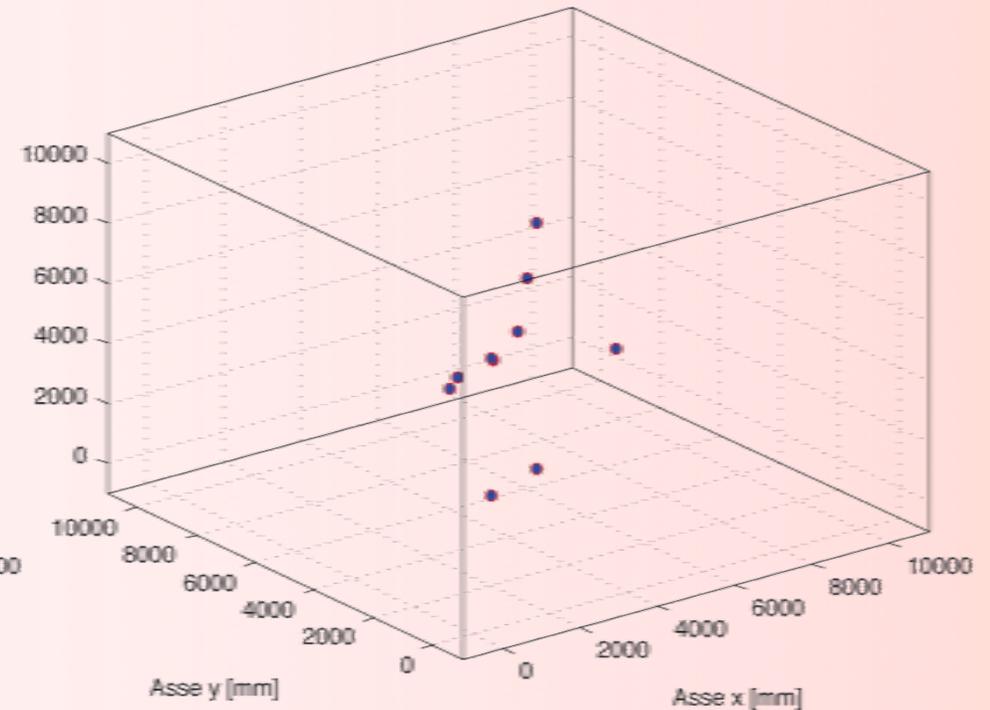
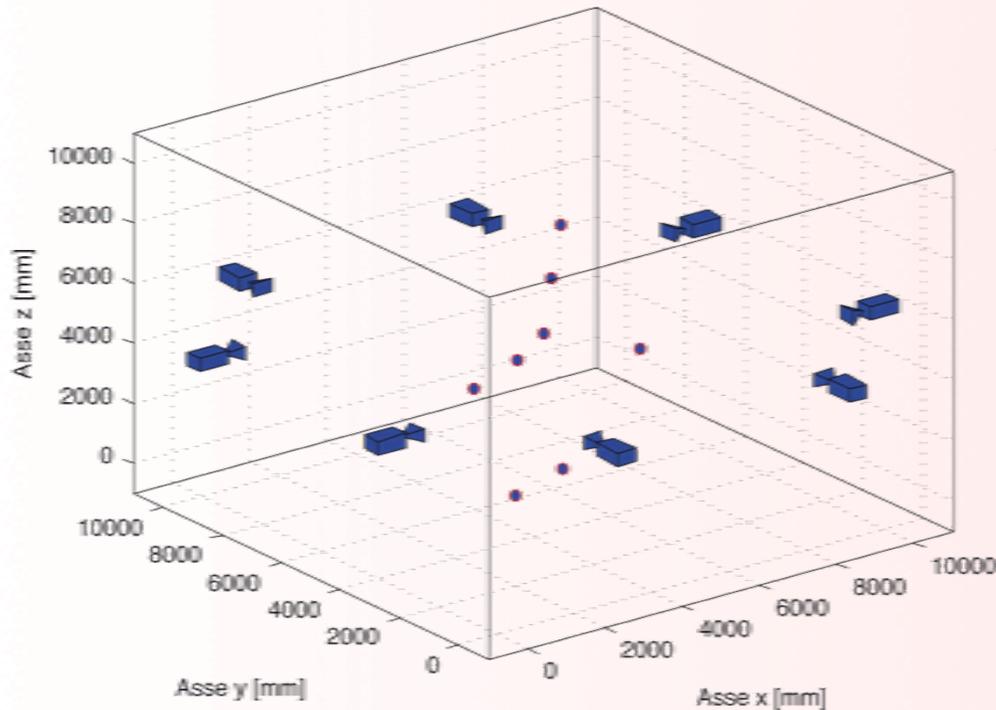
Algoritmo Centralizzato

Ingressi

- Coordinate centri nel sistema di riferimento assoluto
- Posizione delle telecamere
- Distanza minima per la ricostruzione

Uscita

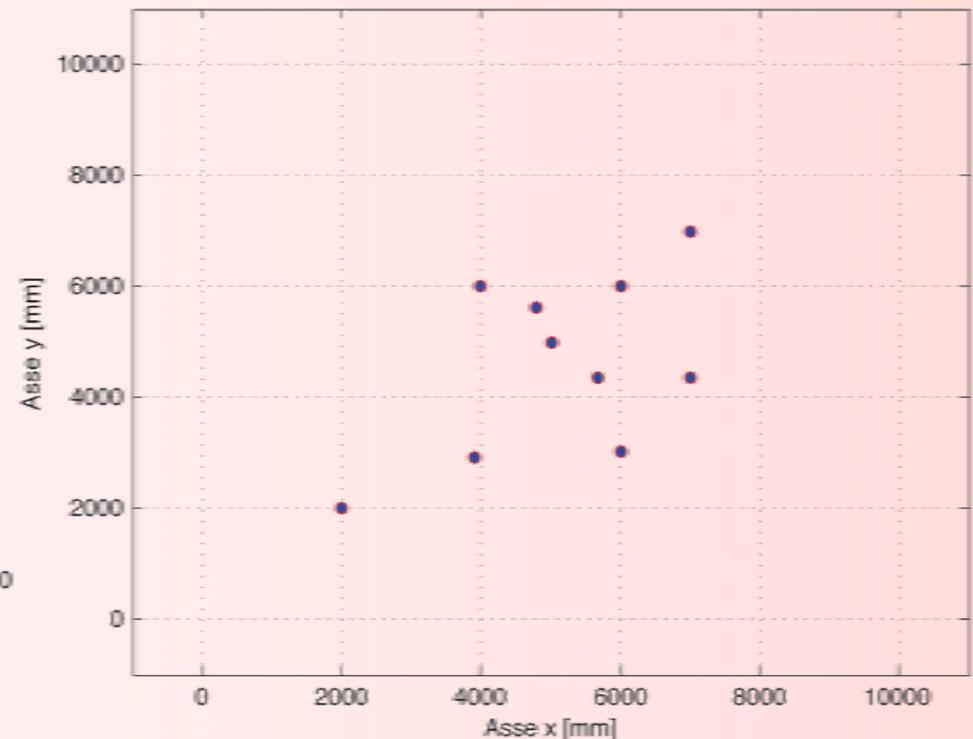
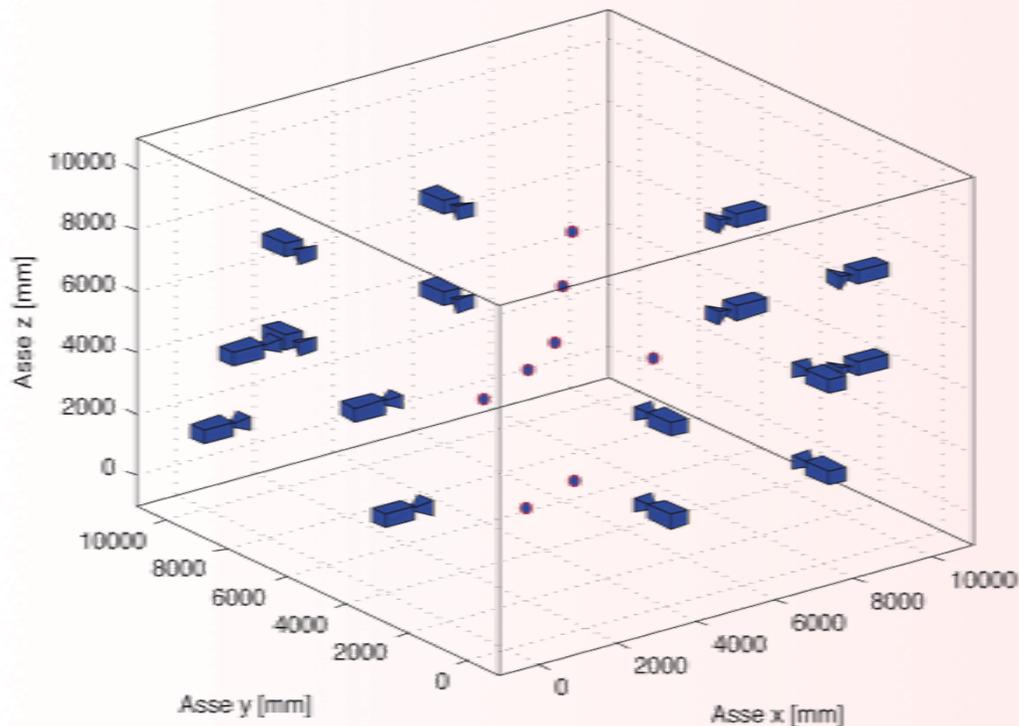
- Marker ricostruiti e distanza



Conclusioni

- Possibilità di ricostruire marker in eccesso
- Possibilità di non ricostruire marker

Algoritmo centralizzato con 16 telecamere



Conclusioni

- Errori di ricostruzione simili
- Variazione dei tempi d'esecuzione

Algoritmo	Tempo d'esecuzione
Distribuito elimina raggi	0,007929 secondi
Distribuito mediazione marker	0,0548 secondi
Distribuito pesato	0,053779 secondi
Centralizzato	0,0809 secondi



Conclusioni e sviluppi futuri

- **Conclusioni:**
 - Minor tempo d'elaborazione rispetto approccio centralizzato
 - Rappresentazione dell'incertezza non soddisfacente
- **Sviluppi futuri:**
 - Utilizzo di più rette per la ricostruzione del marker
 - Implementazione algoritmo ai minimi quadrati
 - Approfondimento sulla rappresentazione dell'incertezza