

Ricostruzione 3D di marker anonimi con distanze reciproche non note

Corrispondenze epipolari, consenso multi-vista e tracciamento

SIMONE TORTORELLA

Ottobre 2026

SOMMARIO

Dopo la calibrazione, i marker sul corpo di un atleta vanno ricostruiti in 3D senza sapere quale blob corrisponde a quale marker e senza conoscere le distanze tra i marker (non sono su un oggetto rigido). Si propone un algoritmo in quattro passi: corrispondenze candidate lungo le linee epipolari tra coppie di camere, conteggio del supporto di ogni candidato nelle altre camere, selezione greedy con il vincolo "un blob, un punto" e affinamento, tracciamento 3D a velocità costante. In simulazione (30 marker da 14 mm su una persona che cammina, 8 camere Full HD, rumore 0,15 px, riflessi) l'algoritmo ricostruisce il 97,0% dei marker visti da almeno due camere con precisione del 99,9% ed errore RMS di 1,14 mm, in 59 ms per frame in Python. Il richiamo dipende soprattutto dal numero di camere (converge tra 8 e 10), l'errore dalla calibrazione e dal rumore dei baricentri: 0,05° di errore di rotazione delle camere raddoppiano l'errore 3D. La tolleranza epipolare ottimale è 1–2 px. Si descrive infine come implementarlo in tempo reale.

Parole chiave: marker anonimi · geometria epipolare · triangolazione multi-vista · corrispondenze · tracciamento 3D

1. Il problema

I Paper 1–3 calibrano le camere con un oggetto rigido di cui si conoscono le distanze tra i marker. Sull'atleta non è così: i marker sono attaccati alla pelle o ai vestiti, le distanze cambiano con il movimento e non sono note in anticipo. Ogni camera vede blob anonimi; alcuni marker sono coperti dal corpo, due marker vicini possono fondersi in un blob, i riflessi aggiungono blob che non corrispondono a nulla. Il compito è trovare le posizioni 3D dei marker usando solo la calibrazione, senza modello del corpo.

La difficoltà è combinatoria: con C camere e n blob per camera le combinazioni di corrispondenze sono n^C . Due raggi di camere diverse si avvicinano anche per caso: un'intersezione di due soli raggi può essere un **punto fantasma**. La geometria epipolare riduce i candidati, il consenso di più camere li conferma.

2. Scena simulata

Una persona cammina (1,4 m/s) o corre (3 m/s) attraverso il volume di 3,5×3,5×2 m. Il corpo è fatto di capsule (tronco, testa, braccia, avambracci, cosce, gambe, piedi) che si muovono con un ciclo di passo; N marker sferici da 14 mm sono distribuiti a caso sulle capsule, fissi rispetto al segmento. Per ogni camera: un marker è visibile se il raggio dalla camera non attraversa nessuna capsula; due marker la cui distanza nell'immagine è inferiore alla somma dei raggi apparenti producono un solo blob nel punto medio; ogni immagine riceve in media 0,3 riflessi; i baricentri hanno rumore gaussiano di 0,15 px. Le camere sono le 8 del Paper 2 (due file di altezza, Full HD, ottiche da 120°). Si simulano 40 frame a 60 fps per ogni configurazione.

3. Algoritmo

Per due camere i, j con rotazione relativa R e traslazione $\mathbf{t}$, un punto 3D visto in coordinate normalizzate $\mathbf{x}_i$ e $\mathbf{x}_j$ (senza distorsione) soddisfa il vincolo epipolare

$$\tilde{\mathbf{x}}_j^\top E \tilde{\mathbf{x}}_i = 0, \quad E = [\mathbf{t}]_\times R \quad (1)$$

La distanza di un blob dalla linea epipolare $\mathbf{l} = E\tilde{\mathbf{x}}_i$ è $|\tilde{\mathbf{x}}_j^\top \mathbf{l}| / \sqrt{l_1^2 + l_2^2}$; moltiplicata per la focale è in pixel e si confronta con la tolleranza ε .

Algoritmo 2 — ricostruzione di marker anonimi (un frame)

1. Togliere la distorsione da tutti i blob (coordinate normalizzate).
 2. Per ogni coppia di camere (i, j) e ogni blob di i : i blob di j entro ε px dalla linea epipolare sono **candidati**; triangolare ogni coppia candidata (DLT).
 3. **Supporto**: proiettare ogni candidato in tutte le altre camere e aggiungere il blob più vicino se entro ε px.
 4. Ordinare i candidati per supporto decrescente e residuo crescente.
 5. Per ogni candidato: togliere le viste i cui blob sono già usati; se restano almeno k viste ($k - 1$ se un tracciato lo prevede vicino), ri-triangolare con tutte le viste (DLT + 3 passi di Gauss-Newton sui pixel), verificare i residui ($< 1,5 \varepsilon$), scartare i duplicati entro 8 mm, marcare i blob come usati.
 6. **Tracciamento**: associare i punti ai tracciati (velocità costante, assegnazione ungherese entro 4 cm); un punto è confermato dopo 2 frame; un tracciato vive 3 frame senza osservazioni.
-

Il costo è dominato dai candidati: $O(C^2 n^2)$ test epipolari e $O(K C)$ proiezioni per il supporto, con K candidati. La regola "un blob, un punto" è quella che elimina i fantasmi: un'intersezione casuale di due raggi raramente trova supporto in una terza camera, e se i suoi blob sono già spiegati da un punto con più supporto viene scartata.

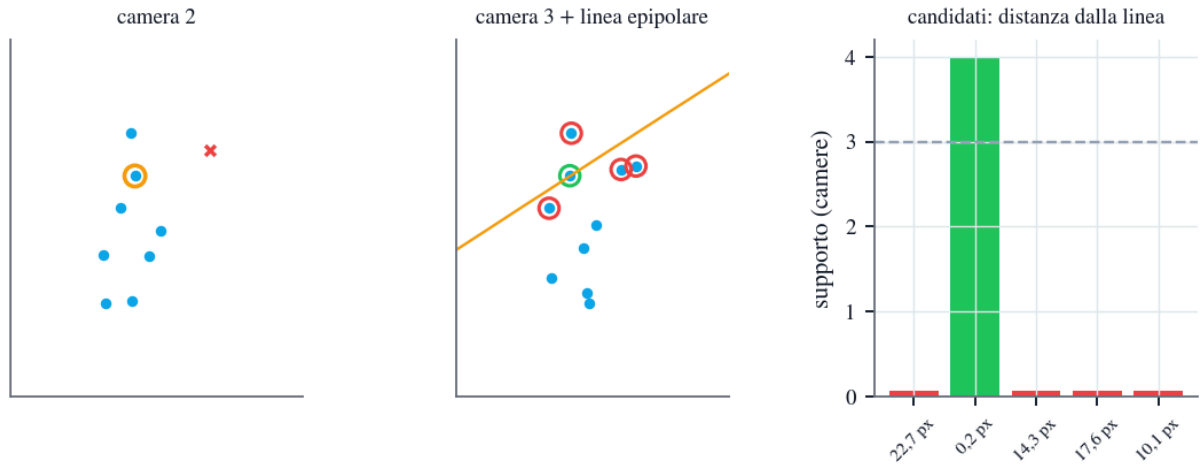


Figura 1. Un frame reale della simulazione. Sinistra: blob della camera A (azzurro: marker; viola: blob fusi; x: riflessi), cerchiato in giallo il blob scelto. Centro: nella camera B la sua linea epipolare e i blob vicini (verde: il marker giusto; rosso: candidati sbagliati). Destra: supporto di ogni candidato nelle 8 camere; tratteggio: supporto minimo 3.

Commento. La linea epipolare attraversa il corpo e passa vicino a parecchi blob: in questo frame 243 candidati in tutto per 24 marker accettati. Solo il candidato giusto è confermato da altre camere; gli altri sono intersezioni casuali che non trovano supporto. È il motivo per cui il supporto minimo 3 elimina i fantasmi.

4. Risultati

Le metriche sono: **richiamo** (quota dei marker visti da almeno 2 camere ricostruiti entro 1 cm), **precisione** (quota dei punti ricostruiti che corrispondono a un marker vero: $1 - \text{fantasmi}$), **errore RMS** dei punti giusti e tempo per frame (Python + NumPy, un core).

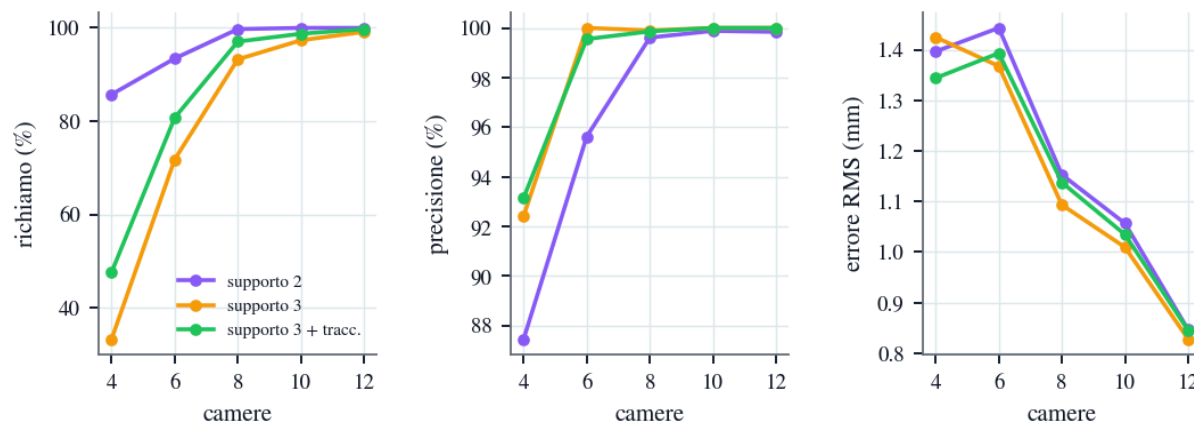


Figura 2. Richiamo, precisione ed errore in funzione del numero di camere, per supporto minimo 2 e 3, con e senza tracciamento (mediana di 2 sequenze).

Commento. Con 4 camere il supporto 2 trova più marker (86%) ma con il 13% di fantasmi; il supporto 3 non ha fantasmi ma trova solo il 33% dei marker. Il tracciamento recupera parte dei marker visti da 2 sole camere quando un tracciato li prevede (48%). Da 8 camere in su le tre varianti convergono: richiamo 97,0%, precisione 99,9%. L'errore scende con il numero di camere ($1,34 \rightarrow 0,85$ mm) perché ogni punto è triangolato con più raggi.

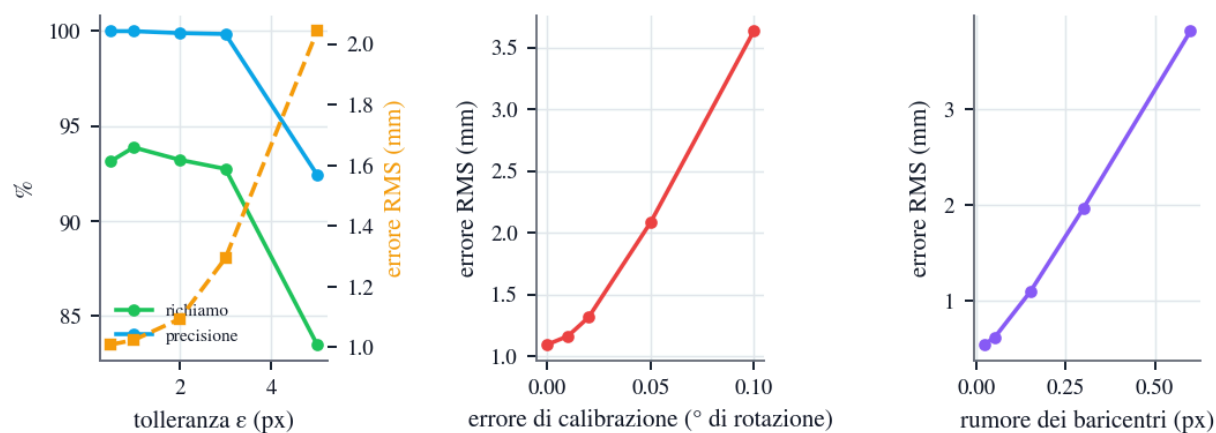


Figura 3. Sinistra: effetto della tolleranza epipolare ϵ . Centro: errore 3D in funzione dell'errore di calibrazione (rotazione casuale di ogni camera; posizione 10 mm per grado). Destra: in funzione del rumore dei baricentri. 8 camere, supporto 3.

Commento. La tolleranza ha un intervallo largo in cui tutto funziona (0,5–3 px); a 5 px compaiono fantasmi (precisione 92%) e l'errore cresce perché si accettano viste sbagliate. $\epsilon = 1-2$ px, cioè circa 5–10 volte il rumore dei baricentri più l'errore di calibrazione proiettato. L'errore 3D segue la calibrazione: con $0,02^\circ$ di rotazione 1,31 mm, con $0,05^\circ$ 2,08 mm, con $0,1^\circ$ 3,63 mm. Il rumore dei baricentri pesa allo stesso modo: da 0,54 mm a 0,02 px a 3,82 mm a 0,6 px. Senza una calibrazione da Paper 1–3, l'errore della ricostruzione sarebbe dominato dalla calibrazione.

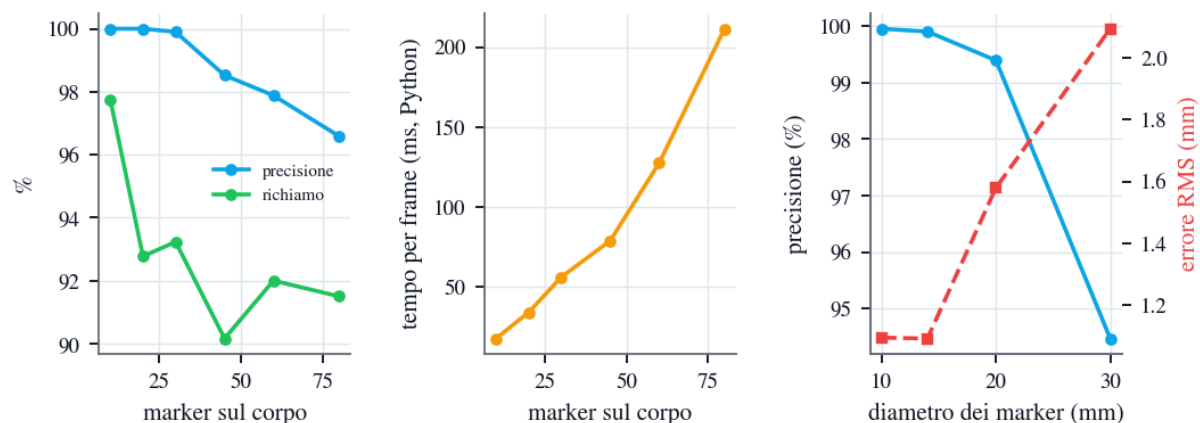


Figura 4. Sinistra e centro: effetto del numero di marker sul corpo. Destra: effetto del diametro dei marker (fusione di blob vicini). 8 camere, supporto 3.

Commento. Più marker significa più blob vicini e più intersezioni casuali: da 30 a 80 marker la precisione scende dal 99,9% al 96,6% e l'errore sale da 1,09 a 1,49 mm; il tempo cresce con il quadrato dei blob per camera (56 → 212 ms). Marker più grandi si fondono più spesso: con 30 mm la precisione scende al 94,5% e l'errore raddoppia (1,09 → 2,09 mm), perché i blob fusi danno punti nel mezzo tra due marker. Sul corpo conviene il marker più piccolo che la risoluzione consente, come per la calibrazione (Paper 1).

Tabella 1. 8 camere. Risoluzione e riflessi: supporto 3 senza tracciamento (rumore fisso di 0,15 px). Cammino e corsa: supporto 3 con tracciamento.

Configurazione	Richiamo	Errore RMS (mm)
HD	92,6	1,67
Full HD	93,2	1,09
4K	93,8	0,61
cammino, 8 camere	97,0	1,14
corsa, 8 camere	88,7	1,23
corsa, 10 camere	89,8	1,10
3 riflessi per immagine	93,4	1,08

La risoluzione riduce l'errore in proporzione ai pixel (il rumore in pixel è fisso): 0,61 mm in 4K contro 1,67 mm in HD. Nella corsa il richiamo scende di qualche punto perché il tracciamento a velocità costante fatica con i piedi (cambi bruschi di velocità) e le occlusioni durano meno ma sono più frequenti; con 10 camere si recupera solo in parte. I riflessi, anche 3 per immagine, non fanno danni: non trovano supporto.

5. Come implementarlo

L'implementazione di riferimento in Python impiega ~59 ms per frame con 8 camere e 30 marker: va bene per l'analisi a posteriori, non per il tempo reale a 120 fps (8,3 ms per frame). Il piano per il tempo reale:

Tabella 2. Moduli del sistema in tempo reale con i costi stimati per 8 camere e 30 marker in C++ (stime da scalare con misure sul prototipo).

Modulo	Dove	Cosa fa	Costo
Blob	camera o box vicino	soglia, componenti connesse, baricentro, diametro, circolarit�; invia solo i blob (≈ 1 kB/frame)	< 1 ms (FPGA/GPU)
Sincronizzazione	server	allinea i frame con il timestamp del trigger (o con i ritardi stimati)	—
Candidati epipolari	server	per ogni coppia ordina i blob lungo le linee epipolari (indice 1D) invece di provarli tutti: $O(C^2 n \log n)$	0,2 ms
Supporto e selezione	server (GPU)	proiezioni vettorializzate, ordinamento, selezione greedy	0,5 ms
Raffinamento	server	DLT + Gauss–Newton 3×3 per punto	0,1 ms
Tracciamento	server	Kalman a velocit� costante, Hungarian 3D	0,1 ms
Etichettatura	server	assegnazione dei punti al modello del corpo (fuori da questo paper)	—

Dalla calibrazione servono intrinseci, distorsione ed estrinseci di ogni camera e, se le camere non hanno trigger hardware, il ritardo stimato (studio preliminare). Durante l'uso, due indicatori dicono se la calibrazione si   degradata (camera urtata, dilatazione termica): il residuo medio dei punti accettati per camera e la quota di blob di ogni camera che non entrano in nessun punto. Se una camera supera le soglie, si ricalibra solo lei con la T.

6. Conclusioni

Sintesi

- Corrispondenze epipolari + supporto in altre camere + regola "un blob, un punto": ricostruisce marker anonimi senza conoscere le distanze e senza modello del corpo.
- 8 camere, supporto 3 e tracciamento: richiamo 97,0%, precisione 99,9%, errore 1,14 mm RMS.
- Parametri: $\epsilon = 1\text{--}2$ px, supporto 3 (2 con tracciato), gate di tracciamento 4 cm (6 cm nella corsa).
- L'accuratezza   decisa dalla calibrazione e dal rumore dei baricentri; il richiamo dal numero di camere.
- Tempo reale: blob sulle camere, candidati con indice epipolare, selezione su GPU: < 2 ms per frame stimati.

Limiti: corpo a capsule, marker distribuiti a caso, nessun modello di pelle che scivola; il passo successivo   l'assegnazione dei punti 3D a un modello del corpo (etichettatura in 3D), che qui non   trattata.

Riferimenti

1. R. Hartley, A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed., 2004 (geometria epipolare, triangolazione).
2. H. W. Kuhn, *The Hungarian method for the assignment problem*, Naval Research Logistics Quarterly 2, 1955.
3. R. E. Kalman, *A new approach to linear filtering and prediction problems*, J. Basic Engineering 82, 1960.
4. M. A. Fischler, R. C. Bolles, *Random sample consensus*, Communications of the ACM 24(6), 1981.

5. S. Tortorella, *Paper 1–3, calibrazione con marker IR anonimi*, 2026.