

Algoritmo per marker anonimi, filtro sulle riproiezioni, numero minimo di frame e criterio di calibrazione riuscita

Dalla bacchetta a T anonima all'errore minimo raggiungibile

SIMONE TORTORELLA

Ottobre 2026

SOMMARIO

Si descrive e si valuta l'algoritmo che calibra 8 camere IR monocromatiche a partire da blob anonimi: riconoscimento della T con fit affine e rifiuto delle ambiguità, stima iniziale dalla terna, bundle adjustment sparso con scarto delle osservazioni a residuo normalizzato alto, ri-etichettatura multi-vista per assegnazione ungherese, bundle adjustment finale. L'etichettatura richiede 0,08 ms per immagine; con soglia di ambiguità 2 lascia l'1,4% di immagini sbagliate, che la ri-etichettatura riduce a meno dello 0,1% delle osservazioni. Lo scarto a $k = 2,5-3$ deviazioni standard è un buon compromesso. Bastano circa 100–200 frame se distribuiti su almeno 20 s di movimento: la durata (copertura del volume) conta più del numero di frame; il bundle adjustment con 400 frame e 8 camere richiede circa 15 s in Python. Il residuo di riproiezione distingue male le calibrazioni buone da quelle sbagliate; una bacchetta di verifica (frame tenuti da parte) va meglio: in 90 sessioni con difetti casuali la regola deviazione della bacchetta $\leq 0,75$ mm, scarto medio $\leq 0,10$ mm e copertura $\geq 5\%$ non accetta mai calibrazioni con p95 oltre 2,9 mm, mentre una regola sul residuo ne accetta fino a 10,6 mm. Il bilancio d'errore indica un minimo raggiungibile di circa 0,10 mm (p95 nel volume coperto, con errore non modellato dei baricentri di 0,02 px) e mostra cosa limita: l'errore non modellato dei baricentri e la tolleranza della terna, non il rumore dei blob.

Parole chiave: bundle adjustment · marker anonimi · riproiezione · rifiuto degli outlier · validazione della calibrazione · bilancio d'errore

1. Introduzione

Il Paper 1 ha fissato l'oggetto (T a 4 marker asimmetrica, terna 3D) e il Paper 2 il numero e la posizione delle camere. Qui si descrive l'algoritmo che trasforma i blob anonimi in una calibrazione, si scelgono le sue soglie con la verità nota della simulazione, si misura quanto costa in frame e in tempo e si cerca un criterio, calcolabile senza conoscere la verità, per dire se una calibrazione è riuscita. Infine si costruisce un bilancio d'errore per capire quale errore minimo si può raggiungere e cosa lo limita.

2. L'algoritmo

Algoritmo 1 — calibrazione da blob anonimi

1. **Blob**: soglia al 40% del bianco, componenti connesse, baricentro pesato, diametro e circolarità di ogni blob.
2. **Terna**: media di 30 frame; PnP per ogni camera con intrinseci nominali → stima iniziale.
3. **Etichettatura per immagine**: per ogni sottoinsieme di 4 blob (anche scartandone uno come riflesso) e ogni permutazione, fit affine al disegno della T; tenere la migliore se $r^* < \tau$ e ogni etichettatura diversa ha $r > \text{amb} \cdot r^*$, altrimenti rifiutare l'immagine.
4. **Filtro di forma**: scartare i blob non circolari o più piccoli di quanto dicono gli altri marker della stessa immagine (occlusione parziale).
5. **Bundle adjustment** sparso (intrinseci, estrinseci, posizioni della T, ritardi) con vincolo di rigidità.
6. **Scarto**: rimuovere le osservazioni con residuo normalizzato $e > k$; ripetere il bundle adjustment (2 volte).
7. **Ri-etichettatura multi-vista**: proiettare in ogni immagine la T stimata da tutte le camere; assegnare *tutti* i blob (anche delle immagini rifiutate) con l'algoritmo ungherese entro g px.
8. **Bundle adjustment finale** con scarto; **verifica** con i frame tenuti da parte (§6).

Il residuo normalizzato di un'osservazione $\mathbf{u}_{ckm}$ (camera c , frame k , marker m) è

$$e_{ckm} = \frac{\|\pi_c(\mathbf{X}_{km} + \delta_c \mathbf{V}_{km}) - \mathbf{u}_{ckm}\|}{\sigma_{ckm}}, \quad \sigma_{ckm}^2 = \sigma_{\text{blob}}^2(d) + \sigma_0^2 \quad (1)$$

dove π_c è la proiezione con distorsione, δ_c il ritardo della camera, $\mathbf{V}$ la velocità del marker e $\sigma_{\text{blob}}(d)$ l'errore atteso del baricentro per un blob di diametro d (Paper 1). La ri-etichettatura risolve, per ogni immagine, l'assegnazione

$$\min_{\phi} \sum_i \|\pi_c(\hat{\mathbf{X}}_{k,\phi(i)}) - \mathbf{u}_{ci}\| \quad \text{s.t. each marker used once, } \|\cdot\| < g \quad (2)$$

3. Etichettatura: soglie e velocità

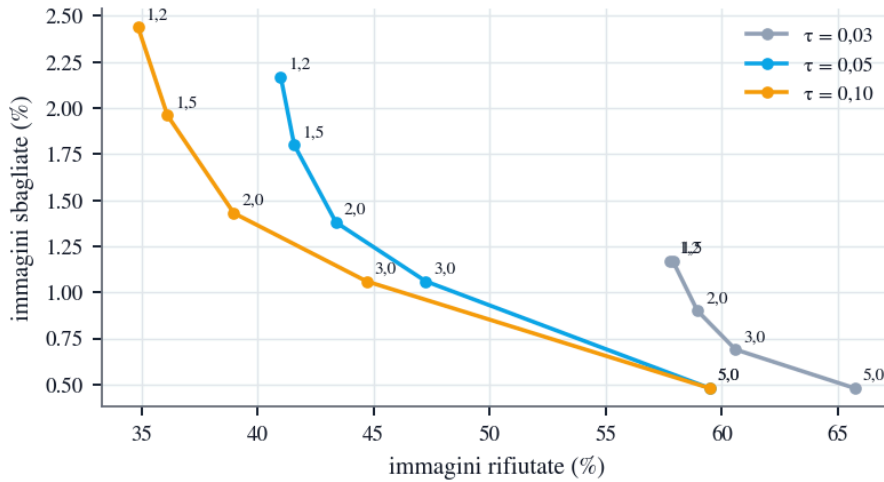


Figura 1. Compromesso tra immagini sbagliate e rifiutate per la soglia di ambiguità amb (numeri sui punti) e la soglia assoluta τ (T4a, 10% di riflessi, 8 camere, 2400 immagini).

Commento. Una soglia di ambiguità più severa riduce gli errori e aumenta i rifiuti; τ sposta la curva. Con $\text{amb} = 2$ e $\tau = 0,05$ si sbaglia l'1,4% delle immagini rifiutandone il 43%. Non serve spingere oltre: le immagini rifiutate vengono recuperate dalla ri-etichettatura e gli errori residui scendono sotto lo 0,1% delle osservazioni (Paper 1). L'etichettatura richiede 0,08 ms per immagine in Python vettorializzato: 8 camere a 60 fps sono gestibili in tempo reale anche senza ottimizzazioni.

4. Riproiezione e soglia di filtraggio

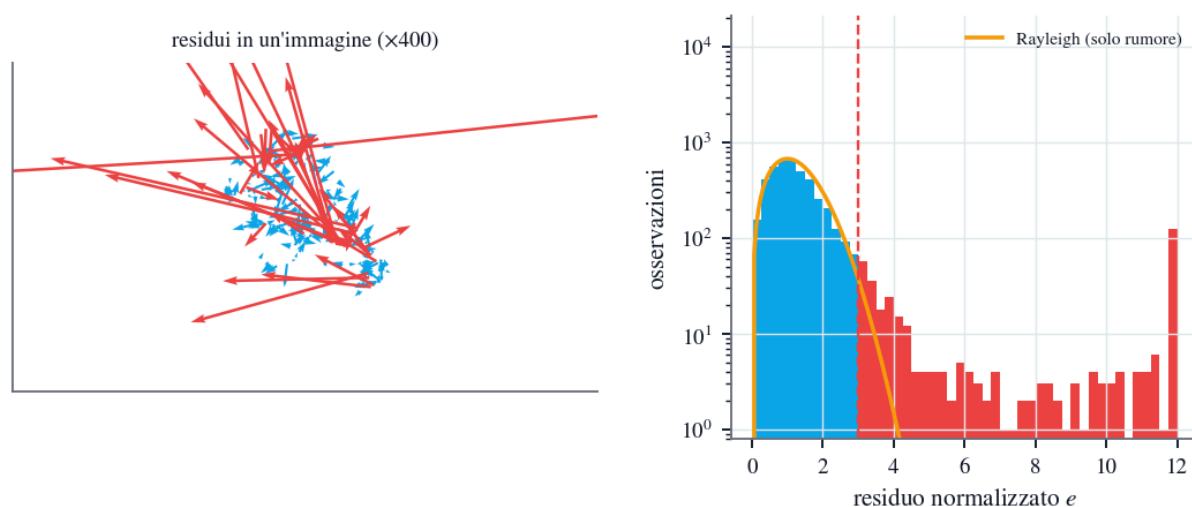


Figura 2. Sinistra: vettori di residuo (ingranditi 400 volte) in un'immagine, rispetto alla soluzione finale; rosso: residuo normalizzato oltre 3. Destra: istogramma dei residui normalizzati di tutte le osservazioni (anche quelle scartate) di una sessione con lo 0,4% di immagini con due marker scambiati; in giallo la distribuzione attesa con solo rumore gaussiano (Rayleigh); l'ultima barra raccoglie i residui oltre 12.

Commento. Con solo rumore gaussiano il residuo normalizzato segue una Rayleigh: oltre 3 ne cadrebbe l'1,1%. Qui oltre 3 cade il 8,4%: oltre alle code del rumore ci sono i marker coperti in parte (il baricentro è spostato di decimi di pixel, Paper 1) e, nell'ultima barra, i marker scambiati, a decine di deviazioni standard. Il filtro a k deviazioni toglie quindi due cose diverse: gli errori grossolani di identificazione e le osservazioni distorte dall'occlusione parziale che il filtro di forma non ha visto.

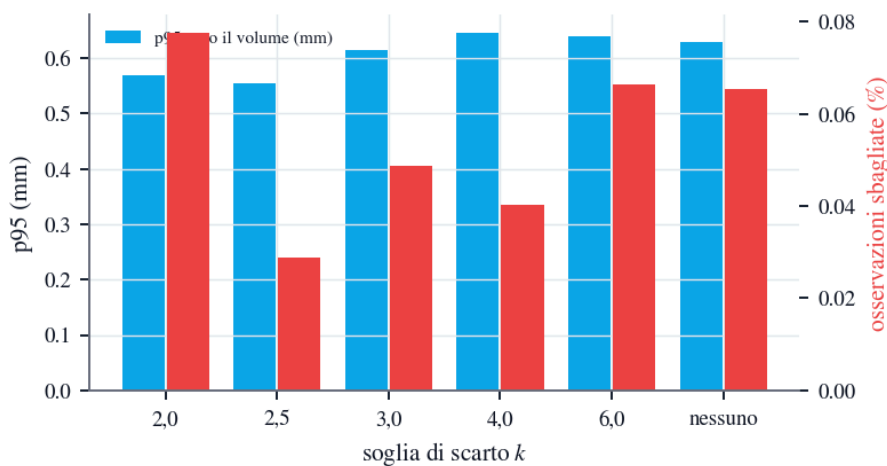


Figura 3. Algoritmo completo (etichettatura, scarto, ri-etichettatura) al variare della soglia di scarto k : errore finale e quota di osservazioni sbagliate rimaste.

Commento. Dopo la ri-etichettatura le osservazioni sbagliate sono così poche (0,03–0,07%) che la soglia cambia poco l'errore finale (0,57 mm a $k = 2$, 0,63 mm senza scarto). Soglie troppo strette ($k = 2$) buttano via il 17% delle osservazioni buone (code della Rayleigh). Si sceglie $k = 2,5$ –3: elimina gli errori grossolani prima della ri-etichettatura, quando sono più numerosi, e costa il 3–8% dei dati.

5. Quanti frame e quanto tempo

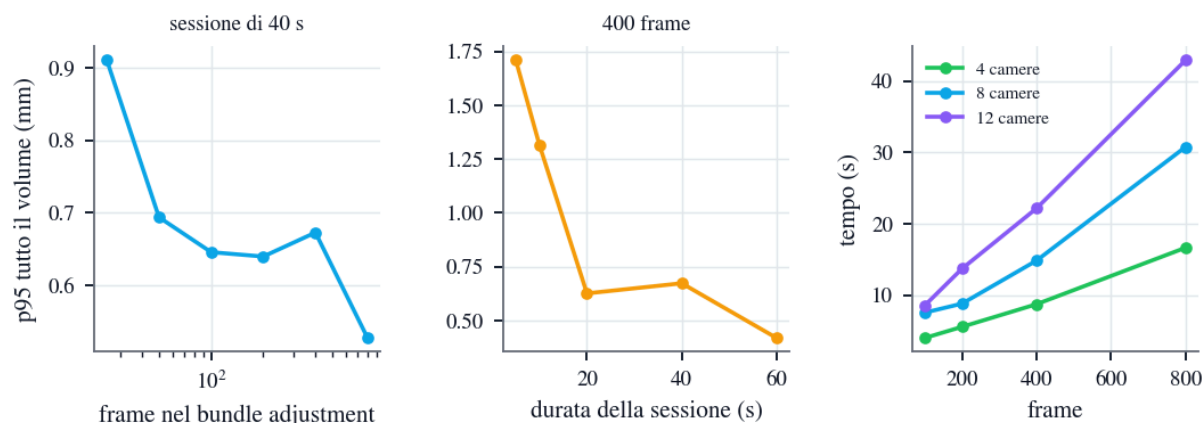


Figura 4. Sinistra: errore in funzione del numero di frame usati (sessione di 40 s). Centro: in funzione della durata della sessione (400 frame). Destra: tempo del bundle adjustment (3 passate, Python + SciPy, un core).

Commento. Con 25 frame l'errore è 0,91 mm; da 100 frame in su si stabilizza (0,65 mm con 100, 0,53 mm con 800). La durata conta di più: a parità di 400 frame, 5 s danno 1,71 mm, 20 s 0,63 mm, 60 s 0,42 mm, perché una sessione lunga copre più volume. **Minimo: 100–200 frame distribuiti su almeno 20 s, meglio 40–60 s.** Il tempo cresce linearmente con i frame e con le camere: 8 camere e 400 frame richiedono circa 15 s; non è un collo di bottiglia, e un'implementazione in C++ (Ceres) sarebbe 10–50 volte più veloce.

6. Quando la calibrazione è riuscita

In una installazione reale la verità non c'è. Servono grandezze calcolabili che predicano l'errore. Si sono simulate 90 sessioni con difetti casuali (pochi frame, sessioni brevi, persona che esce dal volume, focale iniziale sbagliata fino al 30%, terna imprecisa, errore non modellato fino a 0,3 px, fino al 5% di immagini con marker scambiati) e per ognuna si sono calcolati: il residuo di riproiezione RMS, la **copertura** (quota del volume entro 30 cm da una posizione della T vista da ≥ 2 camere), il numero minimo di osservazioni per camera e una **bacchetta di verifica**: un frame su cinque non entra nel bundle adjustment e viene triangolato con le camere stimate; si confrontano le distanze tra i marker con quelle note della T (media e deviazione standard dello scarto). Una sessione è "buona" se il p95 nel volume coperto è ≤ 1 mm.

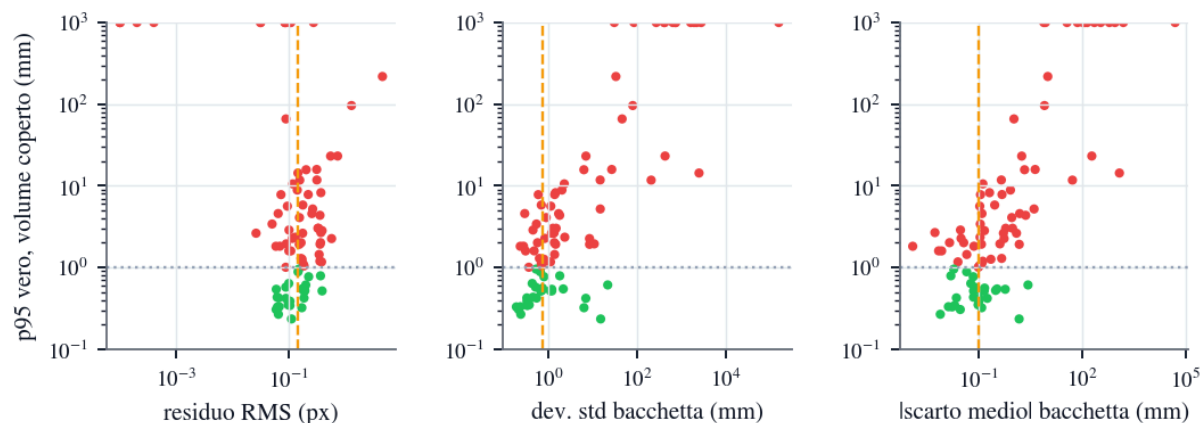


Figura 5. 90 sessioni con difetti casuali: p95 vero (volume coperto) in funzione di tre diagnostiche. Verde: buona (≤ 1 mm); rosso: no. Tratteggio giallo: soglie delle regole.

Commento. Nessuna diagnostica separa perfettamente le due classi, ma alcune molto meglio di altre. Il residuo di ri-proiezione è la peggiore (AUC 0,60: 0,5 sarebbe il caso, 1 la separazione perfetta); con pochi dati il bundle adjustment adatta camere e punti alle osservazioni e il residuo resta piccolo anche quando la calibrazione è sbagliata. La bacchetta di verifica è la migliore (AUC 0,76 per la deviazione standard, 0,76 per lo scarto medio), seguita dal numero di osservazioni per camera (0,73).

Tabella 1. Le due regole sulle 90 sessioni (27 buone).

Regola	accetta- te	di cui buone	p95 massimo tra le accettate
bacchetta: dev. $\leq 0,75$ mm, $ media \leq 0,10$ mm, copertura $\geq 5\%$	20	13	2,9 mm
residuo $\leq 0,15$ px, copertura $\geq 5\%$	29	14	10,6 mm

La regola sulla bacchetta accetta anche qualche calibrazione sopra 1 mm, ma **mai oltre 2,9 mm**; la regola sul residuo ne accetta fino a 10,6 mm. In pratica la bacchetta va usata come stima dell'errore più che come test sì/no: il 95° percentile degli scarti della bacchetta ha correlazione 0,70 (in scala logaritmica) con l'errore vero, il residuo 0,49. Un controllo indipendente è ancora meglio: una seconda bacchetta di lunghezza nota mossa in tutto il volume dopo la calibrazione, perché i frame di verifica qui vengono dalla stessa traiettoria della calibrazione e non vedono le zone che la T non ha attraversato.

7. Bilancio d'errore e fattibilità

Per capire quale errore minimo si può raggiungere si parte dalla configurazione di riferimento (8 camere Full HD a due altezze, 60 fps, 20 s, marker da 30 mm, trigger hardware) e si toglie o si migliora una sorgente d'errore alla volta.

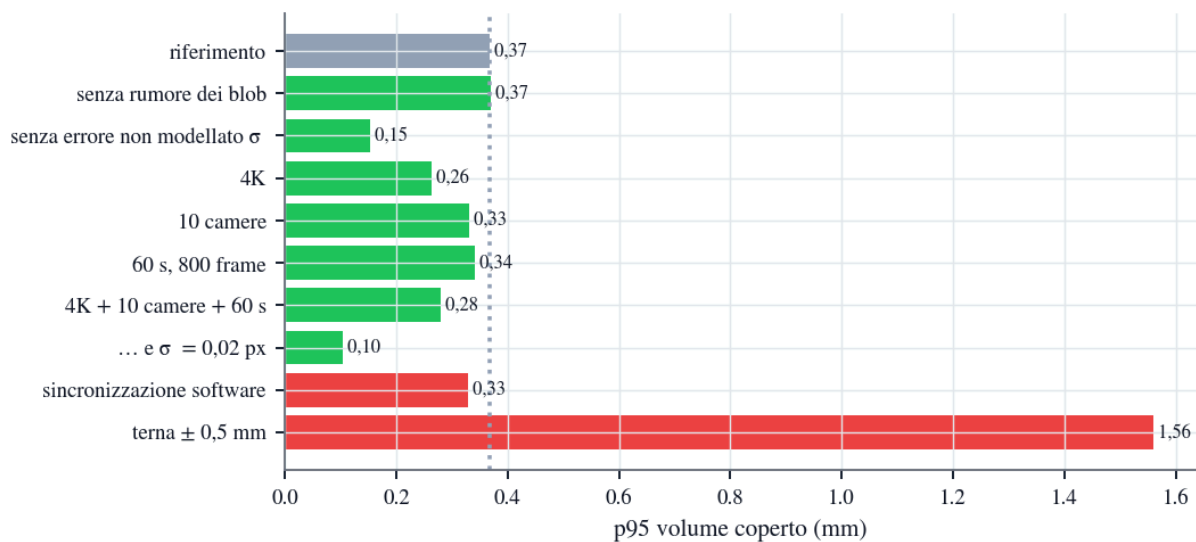


Figura 6. Bilancio d'errore: p95 nel volume coperto togliendo o migliorando una sorgente alla volta (mediana di 4 sessioni). Verde: miglioramenti; rosso: peggioramenti.

Commento. Il rumore dei blob (modello del Paper 1) è irrilevante: toglierlo non cambia nulla (0,37 contro 0,37 mm). Pesa l'errore non modellato σ_0 : senza, si scende a 0,15 mm. 4K, 10 camere e sessioni di 60 s aiutano poco singolarmente (0,26, 0,33, 0,34 mm); tutti insieme portano a 0,28 mm con $\sigma_0 = 0,08$ px e a **0,10 mm con $\sigma_0 = 0,02$ px, il minimo raggiungibile in questo scenario** (p95 nel volume coperto; 0,14 mm sul volume intero). Due sorgenti invece lo rovinano: la sincronizzazione software (0,33 mm sul volume coperto, 3,01 mm sul volume intero) e la tolleranza della terna (1,56 mm con $\pm 0,5$ mm).

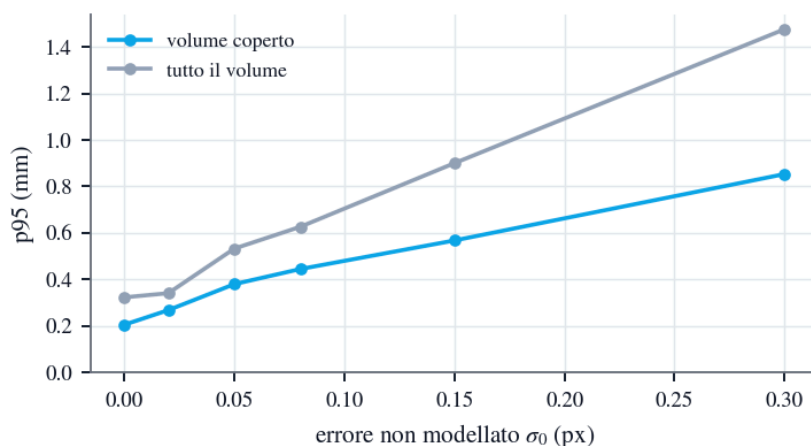


Figura 7. Sensibilità all'errore non modellato dei baricentri σ_0 (8 camere Full HD).

Commento. L'errore cresce quasi linearmente con σ_0 : da 0,27 mm a 0,02 px a 0,85 mm a 0,3 px. È il parametro da misurare per primo su un prototipo, fotografando una sfera ferma per qualche minuto: la deviazione standard del baricentro, dopo aver tolto vibrazioni lente, è una stima diretta di σ_0 .

Tabella 2. Fattibilità: cosa limita l'errore e cosa si può fare per abbassarlo ulteriormente.

Parametro	Effetto	Come abbassarlo
Errore non modellato σ_0	dominante; $\sim$ lineare	modello di distorsione più ricco, global shutter, montaggi rigidi, calibrazione termica
Tolleranza della terna	dominante sull'assoluto	terna lavorata a 0,1 mm o misurata con braccio di misura
Sincronizzazione	$\times 4-6$ sul volume	trigger hardware
Occlusione parziale	bias $\propto$ diametro	marker piccoli, filtro di forma, scarto a 3σ
Copertura del volume	errore fuori dal volume coperto	sessioni di 40–60 s guidate da una mappa di copertura in tempo reale
Risoluzione / numero di camere	secondario per la calibrazione	4K se i marker devono essere piccoli

8. Conclusioni

Algoritmo, soglie, errore minimo

- Etichettatura: fit affine su tutte le assegnazioni, $\text{amb} = 2$, $\tau = 0,05$; nessun tracciamento (propaga gli errori).
- Filtri: forma (circularità $\geq 0,93$, diametro $\geq 0,9 \times$ quello atteso) e scarto a $k = 2,5-3$ del residuo normalizzato.
- Ri-etichettatura multi-vista entro 6 px (Full HD) prima del bundle adjustment finale.
- Sessione: ≥ 20 s (meglio 40–60 s), 100–400 frame; T vista da $\geq 2-3$ camere insieme.
- Criterio: bacchetta di verifica con deviazione $\leq 0,75$ mm e scarto medio $\leq 0,10$ mm, copertura $\geq 5\%$ (mai accettate calibrazioni oltre 2,9 mm); meglio ancora una seconda bacchetta mossa in tutto il volume. Il residuo di riproiezione da solo non basta.
- Errore minimo raggiungibile in simulazione $\sim 0,10$ mm (p95, volume coperto); in pratica limitato da σ_0 e dalla terna.

Limiti: le soglie sono state scelte su dati simulati con la verità nota; su un sistema reale vanno verificate con un oggetto di misura indipendente (ad esempio una seconda bacchetta di lunghezza nota). Le 90 sessioni del criterio coprono difetti plausibili ma non tutti (per esempio camere urtate durante la sessione).

Riferimenti

1. B. Triggs, P. McLauchlan, R. Hartley, A. Fitzgibbon, *Bundle adjustment — a modern synthesis*, LNCS 1883, 2000.
2. H. W. Kuhn, *The Hungarian method for the assignment problem*, Naval Research Logistics Quarterly 2, 1955.
3. R. Hartley, A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed., 2004.
4. M. A. Fischler, R. C. Bolles, *Random sample consensus*, Communications of the ACM 24(6), 1981.
5. W. Kabsch, *A solution for the best rotation to relate two sets of vectors*, Acta Crystallographica A32, 1976.