Consiglio Nazionale delle Ricerche
Istituto di Calcolo e Reti ad Alte Prestazioni

Un sistema di visione stereoscopica per un robot mobile

I. Infantino, P. Cancelliere, E. Morello, A. Termine

Rapporto Tecnico N.:
RT-ICAR-PA-06-06 **luglio 2006**

Consiglio Nazionale delle Ricerche, Istituto di Calcolo e Reti ad Alte Prestazioni (ICAR)
– Sede di Cosenza, Via P. Bucci 41C, 87036 Rende, Italy, URL: *www.icar.cnr.it*
– Sede di Napoli, Via P. Castellino 111, 80131 Napoli, URL: *www.na.icar.cnr.it*
– Sede di Palermo, Viale delle Scienze, 90128 Palermo, URL: *www.pa.icar.cnr.it*

Consiglio Nazionale delle Ricerche
Istituto di Calcolo e Reti ad Alte Prestazioni

Un sistema di visione stereoscopica per un robot mobile

I. Infantino [1], P. Cancelliere [2], E. Morello [2], A. Termine [2]

Rapporto Tecnico N.:
RT-ICAR-PA-06-06

Data:
luglio 2006

---

[1] Istituto di Calcolo e Reti ad Alte Prestazioni, ICAR-CNR, Sede di Palermo Viale delle Scienze edificio 11 90128 Palermo
[2] Università degli Studi di Palermo Dipartimento di Ingegneria Informatica Viale delle Scienze 90128 Palermo

SOMMARIO

INDICE FIGURE

1. Introduzione

In questo lavoro sono state affrontate le problematiche relative all'implementazione di una piattaforma mobile provvista di visione stereoscopica. L'obiettivo principale in questo progetto è quello di permettere al Robot di individuare un oggetto mobile e una volta localizzato di avvicinarsi ad esso. Naturalmente il sistema per poter svolgere tale lavoro deve essere in grado di determinare con precisione la posizione relativa del suo obiettivo e ciò è possibile grazie all'ausilio del sistema di visione stereo integrato. Per l'implementazione degli algoritmi si è tenuto conto della loro complessità ed efficienza al fine di rendere tutto il processo elaborativo real-time. Per il motivo appena detto in fase di progettazione è stato pensato di suddividere le elaborazioni in due blocchi principali, uno riguardante i calcoli di supporto alla visione stereo che possono essere eseguiti off-line e uno relativo all'intero processo di percezione-azione da eseguire in real-time. Di seguito verrà illustrata l'architettura della piattaforma mobile e il sistema software che la controlla.

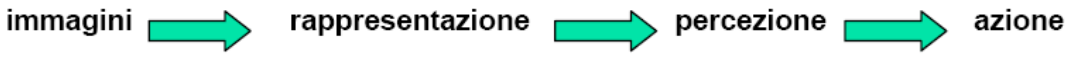
2. Visione Stereoscopica

2.1 Introduzione alla visione stereoscopica

Nel settore della navigazione autonoma, uno dei maggiori problemi aperti consiste nella limitata capacità dei robot di percepire l'ambiente circostante e di reagire con sufficiente tempestività. L'individuazione d'ostacoli e la stima della posizione sono problemi presenti in tutte le applicazioni che richiedano l'uso di robot in grado di muoversi autonomamente, senza essere teleguidati.

Tra i molteplici processi percettivi di cui un robot può avvalersi, la visione costituisce quello che fornisce il maggior numero di informazioni, sia in termini qualitativi che quantitativi. La visione artificiale è il processo che estrae informazioni da una scena analizzando l'immagine della scena stessa.

L'idea alla base del processo di visione consiste nell'avere una o più telecamere connesse ad un Robot, il quale deve automaticamente interpretare le immagini di una scena reale, ottenendo informazioni per svolgere determinate azioni.



I sensori preposti alla percezione visiva, le telecamere, sono dotati di un dispositivo ottico in grado di formare un'immagine della scena su un piano: in esso, per ogni punto, la luminosità ed il colore sono correlati alla luce ricevuta dal sensore e proveniente da una certa direzione che è la luce riflessa da un punto di un oggetto in quella direzione.

Per poter essere sottoposta ad elaborazione, l'immagine deve essere discretizzata, ottenendo una matrice di valori, i pixel, ognuno legato alla luminosità o al colore in quel punto o alla media in un intorno. L'immagine o una sequenza di immagini della scena costituiscono l'input dell'intero processo e le informazioni estratte ne costituiscono l'output. Nello specifico l'input è

dato da una coppia di immagini relative alla stessa scena, ma acquisite da due angolazioni diverse. Si parlerà allora di visione stereoscopica.

Il cervello umano è in grado di fondere le immagini retiniche che provengono dagli occhi e di percepirle come un'immagine unica.

È proprio quest'ultimo processo, definito Stereopsi, che ci consente la localizzazione relativa degli oggetti visivi in profondità, donandoci la percezione della tridimensionalità dello spazio.

La stereoscopia permette quindi di trarre informazioni utili per la navigazione, la manipolazione ed il riconoscimento.

L'obiettivo da perseguire è quindi quello di riconoscere gli oggetti presenti nella scena e determinarne la posizione nel mondo rispetto ad un riferimento, che è il robot stesso. In linea di principio, la tipologia delle informazioni estratte dipende dall'applicazione e dalle sue finalità ed esigenze.

L'approccio stereoscopico pone alcune ingenti problematiche connesse alla realizzazione pratica del sistema di stereo visione, che impone l'impiego di due telecamere perfettamente uguali e ben allineate sullo stesso asse, l'uso di due sistemi di acquisizione, in grado di lavorare parallelamente, e la conoscenza esatta dei parametri di calibrazione dei dispositivi.

Infatti, sebbene le leggi prospettiche che legano le coordinate 3D dei punti dello spazio a quelle delle rispettive proiezioni sui piani immagine sono note, la possibilità di utilizzare quelle relazioni geometriche è comunque subordinata alla conoscenza esatta dei parametri interni delle telecamere utilizzate, e alla conoscenza quasi perfetta della disposizione geometrica dei sensori (parametri esterni).

Grazie allo studio fatto, le difficoltà relative all'hardware sono state finalmente superate, mentre lo studio di algoritmi di ottimizzazione e risoluzione di funzioni non lineari complesse ha consentito di misurare con un grado di accuratezza notevole i parametri intrinseci ed estrinseci delle telecamere correggendo via software le non idealità del sistema.

Per merito di ciò il sistema di visione stereoscopica utilizzato dal robot mobile permette il riconoscimento di oggetti e l'esplorazione di ambienti.

I vantaggi della visione stereoscopica risiedono infatti nel recupero della profondità, che non solo permette di ricostruire la tridimensionalità degli oggetti presenti sulla scena, ma anche di percepire la vicinanza di ostacoli o dedurre l'avvicinamento o allontanamento di corpi in moto, e infine di risalire all'intera struttura 3D di una scena.

2.2 Acquisizione delle immagini

Dal punto di vista dell'informazione, la visione implica una notevole riduzione dei dati a disposizione; pertanto prima di approdare al riconoscimento effettivo degli oggetti, occorre illustrare le fasi di elaborazione fondamentali (preprocessing). Successivamente verrà descritta la strategia implementata ai fini dell'individuazione dell'oggetto.

Il preprocessing di un'immagine costituisce un'elaborazione di basso livello che consente di ridurre notevolmente la quantità di informazione, limitandola in tal modo a quella effettivamente utile ai fini del riconoscimento, e diminuendo così ulteriormente il tempo di elaborazione. Il sistema di visione acquisisce immagini a colori delle dimensioni di 320x240 pixel in formato RGB, il che equivale a triplicare il numero di punti disponibili portandolo a 320x240x3=230400 punti o byte (ogni componente cromatica è memorizzata in formato ad 8 bit). Nell'ipotesi in cui il sistema di visione acquisisca immagini con un frame-rate nelle condizioni di massimo regime dichiarato di 15 frame/sec, ciò comporta un flusso di informazione di 3.37 Mbyte/sec.

Ai fini dell'implementazione per il riconoscimento, l' algoritmo di localizzazione:

- deve resistere all'eventuale occlusione di una porzione di superficie, almeno fino al 75% di essa;
- non deve soffrire di problemi di scala dovuti all'allontanamento o all'avvicinamento dell'oggetto;
- deve essere robusto alla vicinanza di altri oggetti che possano essere causa di ambiguità;

- deve poter operare in condizioni di bassa luminosità (illuminazione artificiale scarsa) o in presenza di luce naturale o anche contro luce;
- deve ignorare i riflessi che la luce genera sulla superficie di essa o quelli dell'oggetto sul pavimento (entrambi causa di ambiguità);
- deve essere computazionalmente veloce, tale da garantire un'elaborazione real-time.

Le elaborazioni necessarie a risolvere i problemi dovuti alle condizioni di luminosità vengono effettuate in hardware dalle stesse telecamera.

2.3 Modello della telecamera e Calibrazione

L'approssimazione fatta per l'ottica del sistema di acquisizione è quella della **lente sottile**, che gode delle seguenti proprietà:

1. i raggi paralleli all'asse ottico incidenti sulla lente vengono rifratti in modo da passare per un punto dell'asse ottico, detto fuoco F_r.
2. i raggi che passano per il centro C della lente restano inalterati.

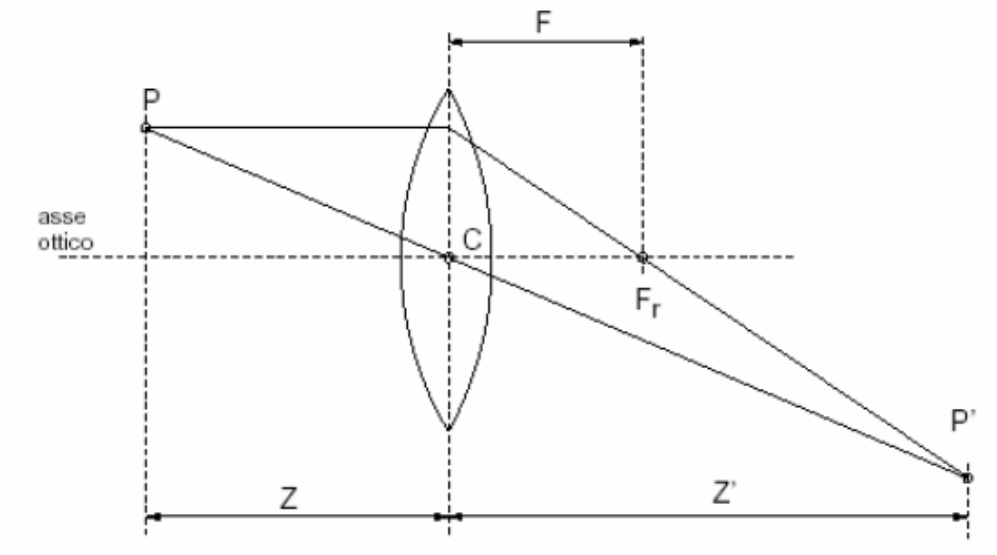


Figura 1 - Modello della lente sottile

Dalle regole geometriche appena citate segue, in virtù dei criteri di similitudine dei triangoli, la legge di Fresnel, che esprime la relazione fra la distanza dell'oggetto z e la distanza immagine z' dal centro della lente, ottenendo in questo modo un modello geometrico della telecamera molto più semplice: il **modello pinhole**.

Il modello pinhole (o prospettico) consiste di un piano retina (o piano immagine) *R,* di un punto *C* , centro, distante f (lunghezza focale) dal piano.

La retta passante per C e ortogonale a R è l'asse ottico (asse z) e la sua intersezione con R prende il nome di punto principale. Il piano F parallelo ad R, e contenente il centro ottico prende il nome di piano focale.

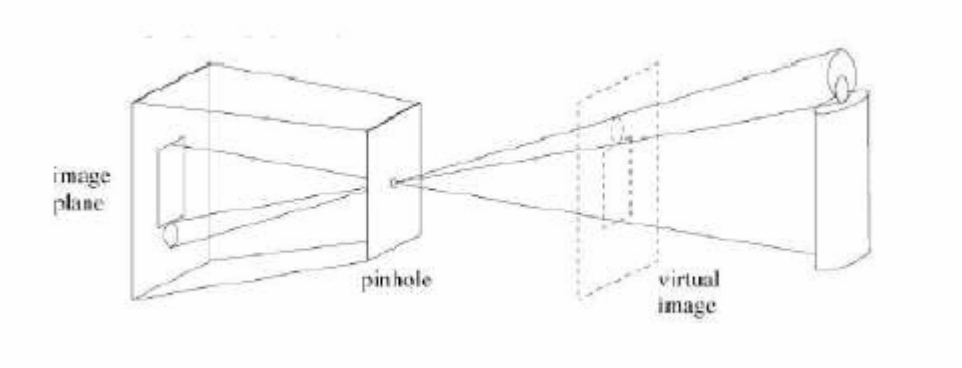


Figura 2 - Modello Pinhole della telecamera

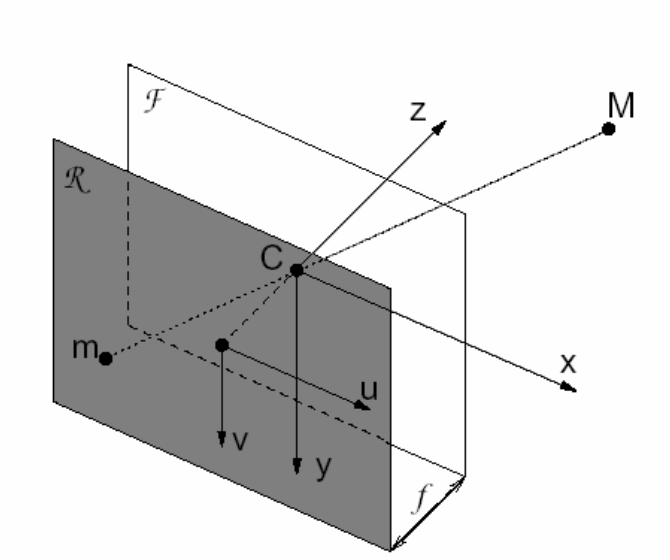


Figura 3 - Modello generale

Il sistema di riferimento cartesiano utilizzato per esprimere le coordinate del punto dello spazio 3D e le coordinate del punto proiettato sul piano immagine è il **Modello generale**, il quale non tiene conto, come nel **modello semplificato**, che il sistema camera coincide con il sistema mondo.

Nel modello utilizzato si considera:

1. la discretizzazione dovuta al sensore CCD (visto come matrice bidimensionale di pixel) e sua posizione rispetto all'asse ottico.
2. la trasformazione isometrica tra il sistema di riferimento mondo e quello della telecamera: **rototraslazione.**

Nella discretizzazione si suppone che:

1. il centro ottico della telecamera non coincide con il centro fisico del CCD ma ha coordinate (u_0, v_0);
2. le coordinate di un punto nel sistema di riferimento standard della telecamera sono misurate in pixel : si introduce quindi un fattore di scala;
3. la forma dei pixel non è quadrata: occorre considerare due fattori di scala diversi, lungo x (k_u), lungo y (k_v).

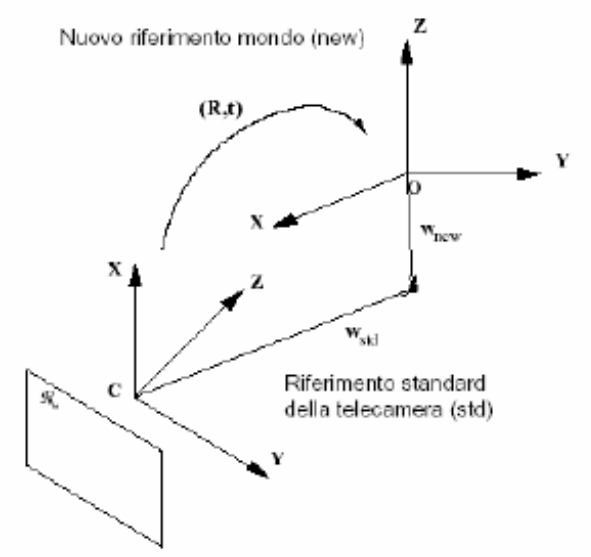
I punti 1, 2, 3 vengono presi in considerazione mediante l'introduzione della traslazione del centro ottico e della riscalatura **indipendente** degli assi u e v:

$$\begin{cases} u = k_u \cdot \dfrac{f}{z} \cdot x + u_0 \\ v = k_v \cdot \dfrac{f}{z} \cdot y + v_0 \end{cases}$$

Quindi la *matrice di calibrazione* [2] A considerata è:

$$P_P = \begin{bmatrix} fk_u & 0 & u_0 & 0 \\ 0 & fk_v & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \qquad A = \begin{bmatrix} fk_u & 0 & u_0 \\ 0 & fk_v & v_0 \\ 0 & 0 & 1 \end{bmatrix}$$

Per tenere conto del fatto che, in generale, il sistema di riferimento mondo non coincide con il sistema di riferimento standard della telecamera, bisogna introdurre la trasformazione rigida che lega i due sistemi di riferimento. Introduciamo dunque un cambio di coordinate costituito da una rotazione R seguita da una traslazione T, ovvero:



$$\begin{bmatrix} X_C \\ Y_C \\ Z_C \\ 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} X_W \\ Y_W \\ Z_W \\ 1 \end{bmatrix}$$

$$P_{RT} = \begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix} \qquad R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \qquad T = \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix}$$

Infine la trasformazione dalle coordinate mondo alle coordinate immagine $\widetilde{m}$ è:

$$\widetilde{m} = \widetilde{P} \cdot w \qquad \text{con} \quad \widetilde{P} = A \cdot [R \mid T] \quad \textit{matrice di proiezione}$$

La matrice A tiene conto dei ***parametri intrinseci*** della telecamera, mentre la matrice P_{RT} dei ***parametri estrinseci***. I parametri indipendenti del modello risultano essere pari a 10, ciò si evidenzia effettuando le seguenti sostituzioni:

$$\alpha_u = fk_u \qquad \alpha_v = fk_v \qquad T = \begin{bmatrix} t_1 \\ t_2 \\ t_3 \end{bmatrix} \qquad R = \begin{bmatrix} r_1^T \\ r_2^T \\ r_3^T \end{bmatrix}$$

La nuova matrice di proiezione prospettica ***MPP*** che tiene conto sia dei parametri intrinseci che di quelli estrinseci è la seguente:

$$P_P = \begin{bmatrix} \alpha_u r_1^T + u_0 r_3^T & \alpha_u t_1 + u_0 t_3 \\ \alpha_v r_2^T + v_0 r_3^T & \alpha_v t_2 + v_0 t_3 \\ r_3^T & t_3 \end{bmatrix}$$

Il modello pinhole della telecamera a 10 parametri che abbiamo fin qui ricavato non è ancora sufficiente a descrivere in maniera completa la trasformazione geometrica che subiscono le immagini acquisite.

Le considerazioni finora fatte riguardo al modello pinhole della telecamera sono state desunte dal modello delle **lenti sottili** esposto all'inizio.

Ma le lenti sottili costituiscono un'astrazione matematica; per quanto i progressi compiuti dall'ottica abbiano portato alla costruzione di lenti sempre più prossime a queste ipotesi di idealità, in esse è sempre presente una certa componente dominante di distorsione **radiale** (dovuta alle dimensioni reali della lente) ed una lieve distorsione **tangenziale** (causata dal decentramento dell'asse della lente).

Ragionando in un sistema di coordinate polari con origine nel centro dell'immagine, la distorsione radiale agisce sui pixel dell'immagine in funzione della sola distanza dal centro **ρ** , mentre quella tangenziale dipende anche dall'angolo ϑ .

Sono stati sviluppati diversi modelli matematici di distorsione al fine di correggerla e tornare così alle ipotesi di lente sottile. Il processo di correzione prende il nome di **compensazione della distorsione**.

Il toolbox utilizzato per la calibrazione effettua tale compensazione applicando il seguente modello:

$$\begin{bmatrix} u_d \\ v_d \end{bmatrix} = \underbrace{(1 + kc_1\rho^2 + kc_2\rho^4 + kc_5\rho^6)}_{\text{Applica la distorsione radiale}} \cdot \begin{bmatrix} u \\ v \end{bmatrix} + \underbrace{\begin{bmatrix} 2kc_3 \cdot u \cdot v + kc_4(\rho^2 + 2u^2) \\ kc_3(\rho^2 + 2v^2) + 2kc_4 \cdot u \cdot v \end{bmatrix}}_{\text{Applica la distorsione tangenziale}}$$

con (u_d, v_d) coordinate distorte. La figura seguente mostra esempi di distorsione con $kc_1 = 0, kc_1 > 0, kc_1 < 0$.

Figura 4 - Esempi di distorsione

2.4 Rettificazione epipolare

Uno dei problemi computazionali della visione stereoscopica riguarda la ricerca dei **punti coniugati o corrispondenti**.

Il calcolo dell'accoppiamento è possibile sfruttando il fatto che le due immagini differiscono solo lievemente, cosicché un particolare della scena appare "simile" nelle due immagini. Il concetto di "similarità" si traduce nella minimizzazione di una funzione costo (tipo correlazione) che scandisce tutte le righe e colonne dell'altra immagine. Basandosi solo su questo vincolo, però, sono possibili molti falsi accoppiamenti ed è inoltre evidente che la ricerca del punto coniugato corrispondente risulta computazionalmente onerosa. In questa sessione introdurremo un vincolo che rende il calcolo delle corrispondenze trattabile: il **vincolo epipolare**, il quale afferma che il corrispondente di un punto in una immagine può trovarsi solo su una retta (**retta epipolare**) nell'altra immagine. Grazie a questo processo, detto **rettificazione epipolare**, la ricerca delle corrispondenze diventa unidimensionale invece che bidimensionale, diminuendo così la complessità computazionale. In seguito verrà introdotto ed esaminato analiticamente il concetto di retta epipolare, ricorrendo all'ausilio della geometria epipolare, e verrà descritta ed analizzata la rettificazione epipolare.

2.4.1 Geometria Epipolare

Vediamo ora quale relazione lega due immagini di una stessa scena ottenute da due telecamere diverse. In particolare, ci chiediamo, dato un punto nella immagine di sinistra, quali vincoli esistono sulla posizione del suo coniugato nella immagine di destra.

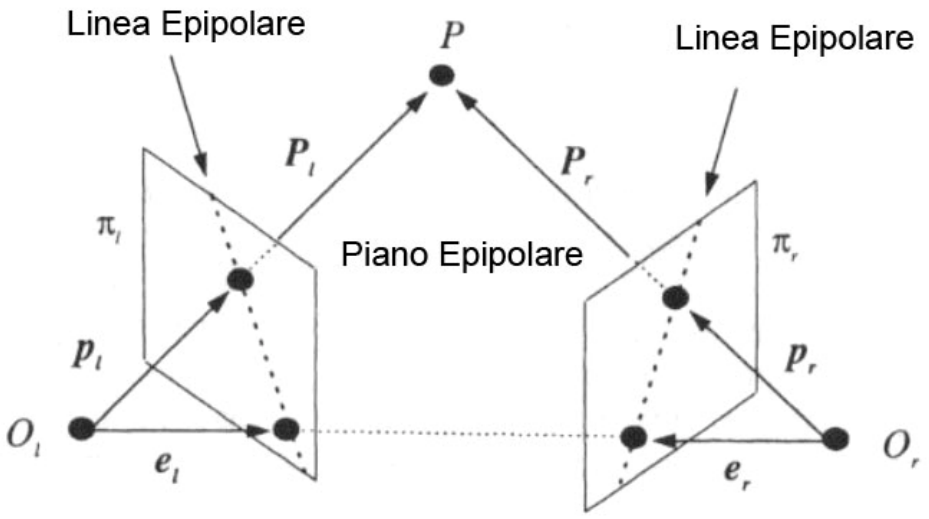


Figura 5 - Epipoli, piani epipolari e rette epipolari.

La scena descritta dalla precedente figura ha le seguenti caratteristiche:

- O_l e O_r sono i centri di proiezione;
- π_l e π_r sono i piani immagine;
- Ogni camera ha un sistema di riferimento con origine in O_l o O_r e asse Z coincidente con l'asse ottico;
- I vettori $P_l = [X_l, Y_l, Z_l]^T$ $P_r = [X_r, Y_r, Z_r]^T$ rappresentano lo stesso punto P nei due sistemi di riferimento;
- I vettori $p_l = [x_l, y_l, z_l]^T$ e $p_r = [x_r, y_r, z_r]^T$ rappresentano le proiezioni di P nei due piani immagine π_l e π_r;

- I punti e_l e e_r in cui la linea congiungente O_l e O_r interseca π_l e π_r sono detti ***epipoli***;

- Il piano $O_l O_r P$ è detto ***piano epipolare*** e interseca ciascuna immagine in una linea chiamata ***linea epipolare***;

La geometria epipolare è importante perché descrive la relazione tra due viste di una stessa scena, dunque è fondamentale in qualunque tecnica di Visione computazionale basata su più di una immagine.
Per determinare il mapping fra i punti coniugati dell'immagine di sinistra e dell'immagine destra, al fine di limitare il problema della ricerca di corrispondenze a un problema 1-D, occorre che le linee epipolari siano, oltre che parallele, orizzontali e collineari. Tale risultato si può ottenere applicando una ***trasformazione omografica***.
Una maggiore trattazione teorica la troviamo negli appunti di "Visione Stereoscopica" di Andrea Fusiello[1].

2.4.2 Determinazione della trasformazione di rettifica

Per effettuare la trasformazione sopra citata, una volta noti i parametri intrinseci ed estrinseci delle telecamere, occorre prima determinare due nuove matrici di proiezione ottenute per rotazione di quelle originarie intorno ai loro centri ottici in maniera tale che i piani focali diventino complanari e contengano la baseline. Inoltre per soddisfare la collinearità tra i punti coniugati le linee epipolari devono avere la stessa coordinata verticale, ovvero le nuove matrici di proiezione devono avere gli stessi parametri intrinseci.

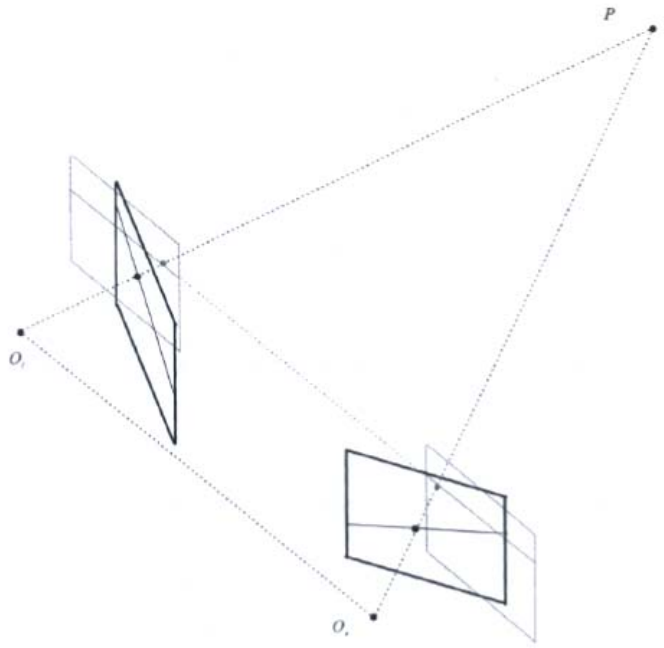


Figura 6 - Piani retina complanari e paralleli alla baseline

Le coordinate dei centri ottici rimangono invariate, viene quindi modificato il sistema di riferimento delle telecamere che possono essere pensate come un'unica telecamera traslata lungo l'asse X. In pratica si effettuano le seguenti operazioni:

- Calcolo della nuova matrice dei parametri intrinseci ponendo $\alpha_u = \alpha_v = \alpha = \min(\alpha_u, \alpha_v)$;
- Nuovo sistema di riferimento con asse X, ovvero la direzione della baseline, uguale a $C_2 - C_1$, asse Y ortogonale al nuovo asse X e asse Z ortogonale ai nuovi assi X e Y;
- Nuova matrice di rotazione Rn individuata dalle componenti della nuova terna di riferimento normalizzate;
- Nuove matrici di traslazione $Tn_1 = -Rn \cdot c_1$ e $Tn_2 = -Rn \cdot c_2$ (espressa in funzione dei centri ottici).

In definitiva le nuove matrici di proiezione prospettica sono:

$$\begin{cases} Pn_1 = An \cdot [Rn \mid Tn_1] \\ Pn_2 = An \cdot [Rn \mid Tn_2] \end{cases}$$

Per determinare le trasformazioni rettificanti in grado di mappare i pixel dell'immagine originaria nel piano immagine finale, ricordando che inizialmente $\tilde{m}_O = \tilde{P} \cdot w$ e quindi $\tilde{m}_f = \tilde{P}n \cdot w$, sostituendo la coordinata mondo esplicitata dalla relazione di partenza otteniamo:

$$\tilde{m} = \tilde{P}n \cdot P^{-1}m \qquad \Rightarrow \qquad \begin{cases} \mathrm{Trf}_1 = \tilde{P}n \cdot \tilde{P}_1^{-1} \\ \mathrm{Trf}_2 = \tilde{P}n \cdot \tilde{P}_2^{-1} \end{cases}$$

2.4.3 Interpolazione

Le matrici *Trf* dette matrici di rettificazione determinano il modo in cui i pixel dell'immagine originaria si mappano in quella rettificata nel nuovo sistema di riferimento. Questa trasformazione introduce delle coordinate non intere, per tale motivo nella corrispondente immagine rettificata possono comparire dei pixel a cui non è stato assegnato nessun valore. La tecnica classica fa uso di opportuni filtri interpolatori non lineari che approssimano il valore cercato.

In questo lavoro si è scelto di utilizzare un'interpolazione bilineare, che approssima il valore cercato in base al valore cromatico dei quattro pixel vicini.

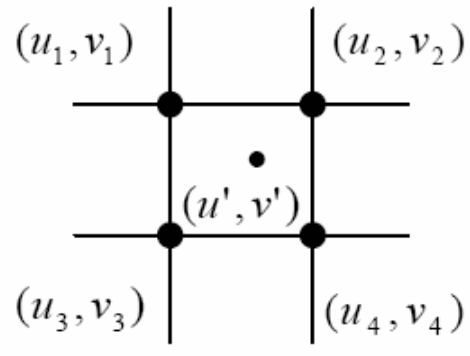


Figura 7 - Interpolazione bilineare

Indicando con V_1, V_2, V_3, V_4 i valori cromatici dei quattro pixel adiacenti a (u', v') vengono calcolati i parametri:

$$dx = u' - u_1$$

$$dy = v' - v_1$$

$$a_1 = (1 - dx) \cdot (1 - dy)$$

$$a_2 = dx \cdot (1 - dy)$$

$$a_3 = (1 - dx) \cdot dy$$

$$a_4 = dx \cdot dy$$

e da questi il valore cromatico $V(u', v') = [a_1\ a_2\ a_3\ a_3] \cdot \begin{bmatrix} V_1 \\ V_2 \\ V_3 \\ V_4 \end{bmatrix}$

3. Architettura hardware del sistema

L'architettura sviluppata per il sistema robotico consta di una unità robotica con a bordo un sistema di visione stereoscopica ed una unità di calcolo ed elaborazione.

L'unità robotica è costituita dal Robot mid-size mobile a sei ruote K-Team "Koala" (320mm x 200mm x 320mm, 3Kg) progettato per le applicazioni del mondo reale con le seguenti caratteristiche:

Processore: Motorola 68331 @ 22MHz ;

RAM: 1Mbyte;

ROM: 1Mbyte;

Motori: 2 servo motori con encoders incrementali integrati che forniscono approssimativamente 19 pulsazioni per mm per il movimento del robot;

Velocità: Max: 0.6 m/s diretta, 0.38 m/s utilizzando il controller PID di velocità;

Min: 0.005 m/s utilizzando il controller PID di velocità

Accelerazione massima: 0.075 m/s² utilizzando il controller PID di velocità;

Sensori:

16 sensori infrarossi di prossimità e luce ambientale;

4 sensori infrarossi opzionali di triangolazione a lungo raggio;

Sensore di temperature ambientale e della batteria;

Alimentazione: Batteria NiMH con memoria di livello di carica

Autonomia (batteria 4Ah): Circa 6 ore (muovendosi continuamente senza carico)

Circa 4 ore (muovendosi continuamente con carico massimo)

Dispositivi I/O disponibili:

12 input digitali [5..12V]

4 output digitali CMOS / TTL

8 uscite digitali alimentate [12V 250mA/output]

6 input analogici (10 bit A/D converter, 4.096v range)

Dimensioni:

Lunghezza: 32 cm

Larghezza: 32 cm

Altezza: 20 cm

Peso: 4kg con la batteria, 3.6kg con il convertitore DC-DC

Figura 8 - Il Robot Koala

L'unità di visione stereoscopica è costituita da una coppia di telecamera "Logitech® QuickCam® Sphere" con le seguenti caratteristiche:

Numero bit/pixel: 32 bit;

Risoluzione massima CCD: 1,3 megapixel;

Modalità rappresentazione colore: RGB;

Frame-rate (max):22 Hz

Lunghezza focale dichiarata 6 mm

Ampiezza angolare campo visivo 45°

Sistema operativo Windows Xp;

Interfaccia di connessione USB 2.0.

Figura 9 - Logitech® QuickCam® Spere

L'unità di calcolo ed elaborazione delle immagini è costituita da un computer portatile di piccolo ingombro, elevate prestazioni e autonomia. Il modello in questione è un "Dialogue Flybook" con display da 8.9'' con le seguenti caratteristiche:

Processore: Transmeta Crusoe TM-5800 a 1 GHz;
Dimensioni: 235 x 155 x 31 mm, 1.230 grammi
Chipset grafico: ATI Radeon Mobility con 16 MB di VRAM con supporto per Microsoft DirectX 9.0
Memoria: 512 MB DDR, 64 KB cache istruzioni 1° livello, 64 KB cache dati 1° livello, 512 KB cache 2° livello.

Figura 10 - L'unità di calcolo ed elaborazione

Figura 11 - Vista posteriore del Sistema

Figura 12 - Vista anteriore del Sistema

Le due telecamera sono fissate, tramite viti, sul corpo macchina del robot in modo da essere orizzontali, parallele e collimate. Quest'ultima ipotesi non può però essere mai verificata nella pratica, neanche per apparati stereoscopici professionali, e sarà dunque necessario effettuare la calibrazione del sistema di visione al fine di avvicinarsi a tali ipotesi di idealità.

Le variabili di controllo del robot sono la velocità di traslazione e la velocità angolare, entrambe riferite al centro del robot (punto medio dell'asse delle ruote). A partire da dette variabili il sistema calcola le velocità da impartire separatamente alle ruote destra e sinistra per generare il moto desiderato.

Il robot utilizza una porta seriale RS232 per comunicare con l'esterno. La comunicazione tra il robot e l'unità di calcolo avviene tramite un adattatore USB-RS232 e mediante l'utilizzo di un hub USB si connettono le unità di visione che compongono il sistema.

4. Architettura software del sistema

Il requisito di funzionalità in real-time ha portato all'individuazione, in fase di progetto, di due fasi di elaborazione distinte. Una fase computazionalmente complessa e comunque statica di elaborazioni off-line in cui vengono determinati i parametri del sistema di visione stereo, l'altra da effettuarsi necessariamente in real-time in cui vengono effettuate le operazioni utili al processo di percezione-azione. L'elaborazione off-line consiste nel determinare i parametri intrinseci ed estrinseci della coppia di telecamere, per poter in seguito essere utilizzati durante l'elaborazione in real-time al fine di effettuare le trasformazioni geometriche del processo di visione.

L'elaborazione real-time consiste nell'acquisizione di un flusso video dal quale si dovranno estrarre le informazioni utili al Robot per la determinazione della posizione e l'inseguimento di un determinato oggetto. Di seguito viene mostrata l'architettura software dell'intero sistema che verrà dopo illustrata in maniera dettagliata.

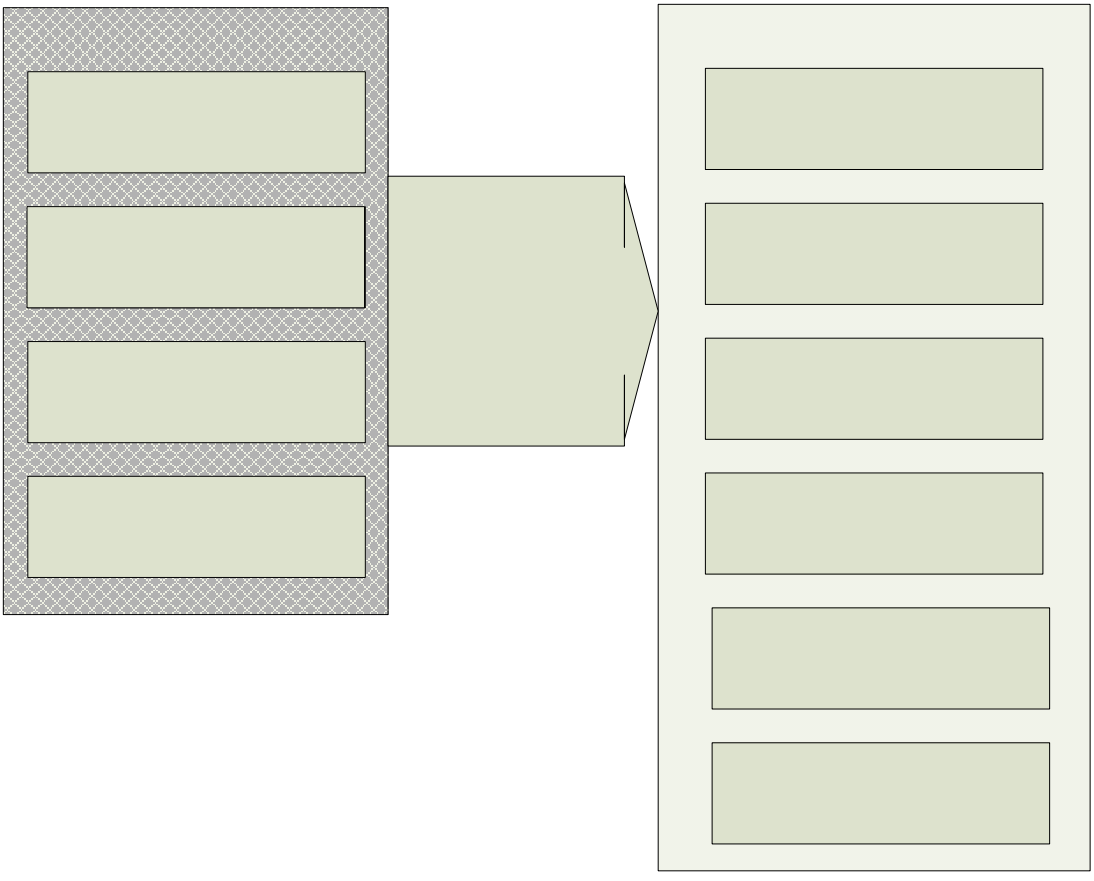


Figura 13 - Architettura software del sistema

4.1 Acquisizione delle immagini per la calibrazione

Prima di effettuare il processo di calibrazione occorre acquisire coppie di immagini stereo adatte a tale scopo. In questo lavoro si è utilizzato un oggetto di calibrazione regolare, planare di dimensioni note, costituito da un pattern tipo scacchiera con quadrati di dimensione 3x3 cm. Il numero dei quadrati della scacchiera deve essere sufficiente a permettere la risoluzione del sistema che porterà alla determinazione dei parametri della telecamera.

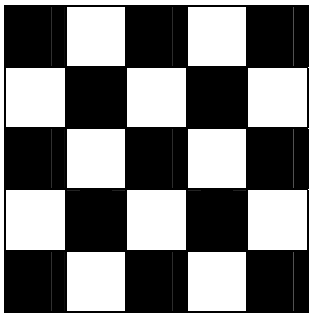


Figura 14 - Esempio di pattern di calibrazione utilizzato

La procedura di calibrazione richiede quindi che vengano acquisite un numero sufficiente di coppie di immagini stereo (non inferiore a 15) con diversa angolazione. Il numero elevato di immagini riprese in diverse angolazioni, per evitare la complanarità dei punti (corner) deve essere tale da permettere, durante la fase di calibrazione, di ridurre al minimo gli errori e l'incertezza associata alla stima dei parametri. Per ottenere le immagini stereo del pattern di calibrazione, ripreso nello stesso istante dalle due telecamere, è stata sviluppata un funzione, presente nell'applicazione Java, che acquisisce quando stabilito la coppia di frame provenienti dal flusso video mostrato nella finestra di anteprima.

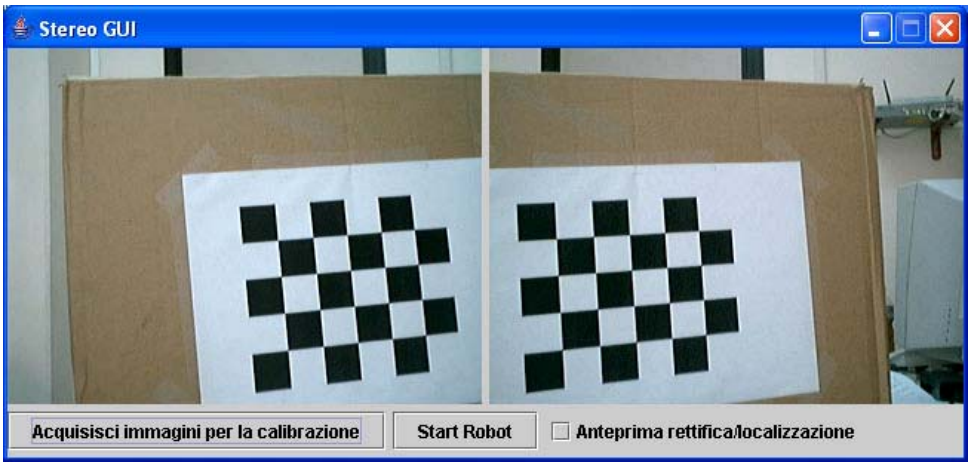


Figura 15 - Funzionalità per l'acquisizione delle immagini di calibrazione

Le immagini vengono quindi salvate sul disco utilizzando un meccanismo di identificazione (immagini di sinistra e di destra) e numerazione dei files necessario per il successivo caricamento da parte del toolbox di calibrazione di Matlab.

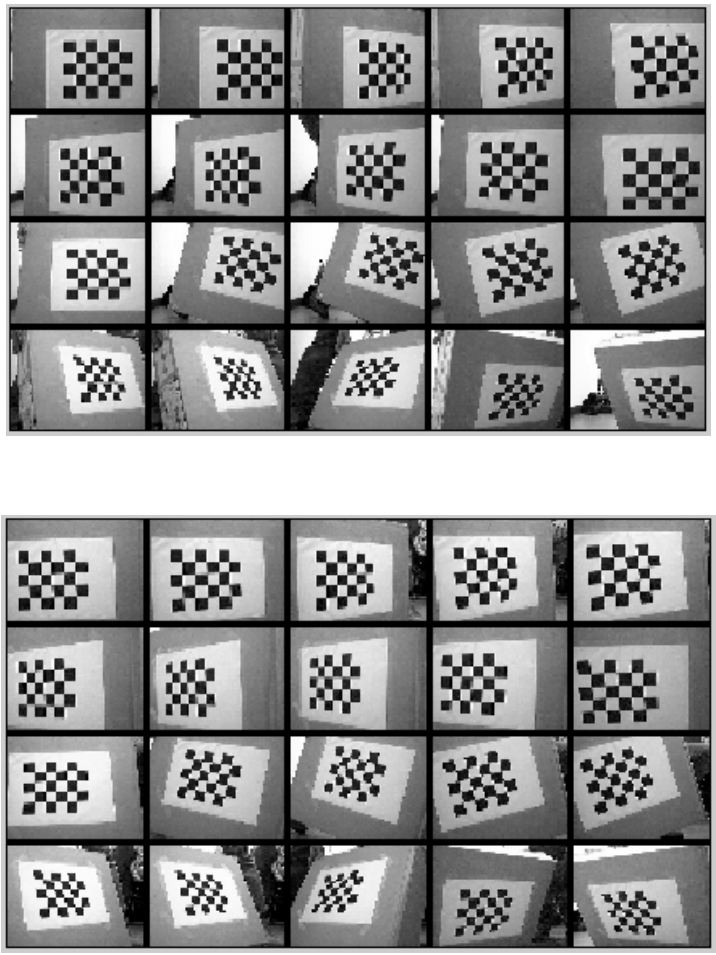


Figura 16 - Esempio di immagini per la calibrazione

4.2 Calibrazione delle telecamere

Il processo di calibrazione visto in teoria è stato eseguito mediante l'ausilio di Matlab attraverso il ***Camera Calibration Toolbox*** [3] sviluppato dalla Microsoft e Intel basato sullo studio di ricerca di Zhang [4]. Il metodo di calibrazione adottato ha la particolarità che non occorre conoscere la posizione assoluta nello spazio dei punti di calibrazione, rispetto ai metodi di calibrazione diretta. Tale processo prevede inizialmente la calibrazione delle singole telecamera e successivamente la calibrazione stereo. L'oggetto di calibrazione utilizzato come già detto consiste in un pattern tipo scacchiera con quadrati di dimensione 3x3 cm.

4.2.1 Procedura di calibrazione stereo

Il processo di calibrazione si articola nelle seguenti fasi:

- Posizionamento relativo delle telecamere sul robot, regolazione della messa a fuoco (via hardware), stima delle posizioni spaziali del pattern, controllo dell'inquadratura;
- Acquisizione delle immagini per la calibrazione;
- Elaborazione delle immagini ed estrazione dei punti di interesse;
- Calcolo dei parametri di calibrazione;
- Analisi degli errori ed eventuali correzioni.

La teoria di calibrazione richiede che tutti i punti estratti dal pattern si trovino sul medesimo piano per cui questo è stato realizzato in maniera tale da garantire una ottima planarità e indeformabilità al variare delle condizioni ambientali ed al trascorrere del tempo.

Prima operazione da effettuare è la scelta della distanza a cui tarare il sistema, questa sarà un compromesso tra le dimensioni della singola inquadratura e la precisione richiesta, ed ovviamente dipenderà dalle dimensioni dell'oggetto da acquisire. In genere è consigliabile che l'oggetto occupi il più possibile la scena inquadrata.

Si passa quindi alla messa a punto delle camere visualizzando a schermo le inquadrature e posizionando le due camere in modo tale che il pattern ricada

al centro dell'immagine per entrambe. Questa orientazione relativa tra le camere rimane invariata per tutta la durata dell'acquisizione.

Finita la messa a punto del layout geometrico del sistema, si passa all'acquisizione delle immagini: si effettuano venti posizionamenti diversi del pattern (in modo da definire un volume di calibrazione) e per ognuna delle venti posizioni si acquisiscono contemporaneamente due immagini, una per la camera destra ed una per quella sinistra. Bisogna assicurarsi che la stessa area sia contenuta interamente in entrambe le immagini riprese dalle camere e che tutte le parti del pattern siano a fuoco anche nelle posizioni angolate.

Tramite le funzionalità del ***Calib_gui*** del toolbox di Matlab si procede alla calibrazione delle singole camere: il software carica in memoria le immagini relative alla singola camera da calibrare ed inizia la loro elaborazione.

Figura 17 - Camera Calibration Toolbox di Matlab

L'estrazione dei punti di calibrazione dalle immagini caricate in memoria si basa su una routine del software che, dopo aver definito le dimensioni in pixel di un puntatore rettangolare, ricerca all'interno dello stesso, la massima variazione del livello di grigio dell'immagine. Le dimensioni del puntatore utilizzato sono di 5x5 pixel. In questo modo, cliccando in maniera approssimata nell'intorno degli spigoli dei quadrati della scacchiera, il sistema individua con ottima precisione la posizione del vertice cercato.

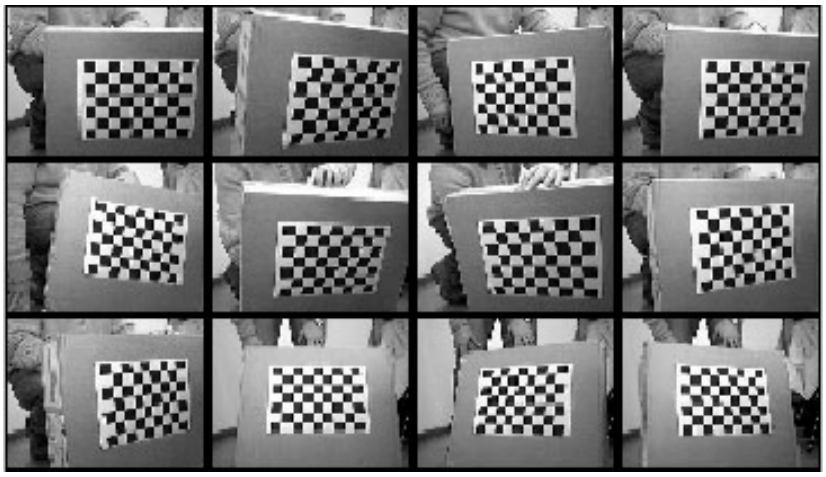


Figura 18 - Immagini del pattern caricate in memoria

Per ognuna delle immagini caricate in memoria è richiesto di ciccare sui quattro spigoli alle estremità della scacchiera definendo così un rettangolo.
L'ordine con cui si clicca per definire i vertici del rettangolo non è casuale, infatti il primo punto su cui si clicca viene riconosciuto come origine del sistema di riferimento associato al pattern e questo è particolarmente importante se bisogna determinare i parametri estrinseci di più camere (cioè determinare la loro posizione reciproca). In questo caso il pattern di punti selezionato deve essere lo stesso in tutte le immagini, così come deve essere lo stesso l'ordine con cui vengono selezionati i punti.

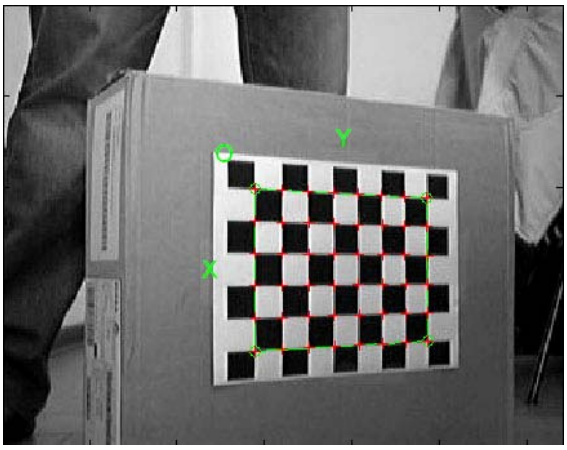


Figura 19 - Estrazione dei corner ed individuazione del sistema di riferimento

Immettendo anche la dimensione del lato dei quadrati stampati sul pattern, il programma individua la posizione teorica di tutti gli spigoli dei quadrati presenti all'interno del rettangolo prima definito e visualizza sullo schermo delle crocette sovrapposte all'immagine reale.

La posizione dei quadrati individuati dal programma all'interno del rettangolo non tiene conto della presenza della distorsione introdotta dalla lente, per cui se gli spigoli trovati si discostano da quelli reali, l'utente può introdurre un valore iniziale della distorsione radiale e ripetere la procedura appena descritta: tale valore può essere modificato finché non si ottiene una corrispondenza soddisfacente.

Dopo l'estrazione dei corner da tutte le immagini, il programma inizia la procedura di calibrazione vera e propria che consiste di due passi: *inizializzazione* e *ottimizzazione non lineare*.

Nella fase di inizializzazione viene calcolato un set di base di parametri di calibrazione, senza considerare la distorsione delle lenti delle camere: i risultati ottenuti in questa fase preliminare vengono utilizzati nell'ottimizzazione non lineare in cui tutti i parametri estrinseci ed intrinseci della camera vengono determinati attraverso un processo che tende a minimizzare iterativamente l'errore di riproiezione (errore tra le coordinate dei punti della scacchiera rispetto al riferimento del pattern e le coordinate dei punti riproiettati, cioè ottenuti utilizzando le matrici di proiezione prospettica prima valutate). I risultati forniti dal programma comprendono la distanza focale, la posizione del punto principale, i fattori della distorsione e l'errore nell'estrazione dei punti; tutte le grandezze sono espresse in pixel ed inoltre è possibile visualizzare su di un grafico l'errore di riproiezione, consentendo così di capire quali sono le immagini in cui l'errore è maggiore (ad ogni immagine è associato un colore diverso): per queste è quindi possibile rieffettuare l'estrazione dei punti ed una nuova ottimizzazione o, nel peggiore dei casi, escluderle dal calcolo.

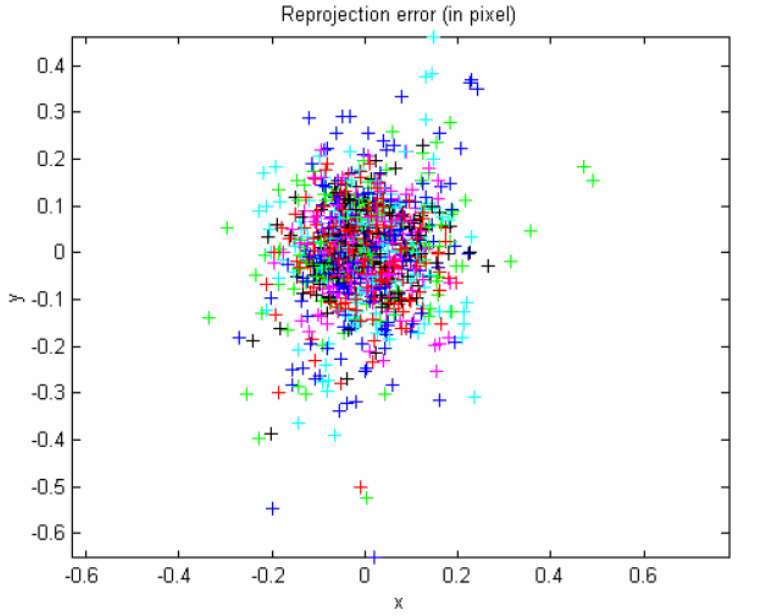


Figura 20 - Errore di riproiezione

Una volta calibrata la prima camera, si procede in maniera del tutto analoga a calibrare la seconda. Ottenuti i valori dei parametri intrinseci delle due camere, viene effettuata un'ulteriore calibrazione, comunemente detta calibrazione stereo, attraverso l'uso delle funzionalità *Stereo_Gui* toolbox di Matlab.

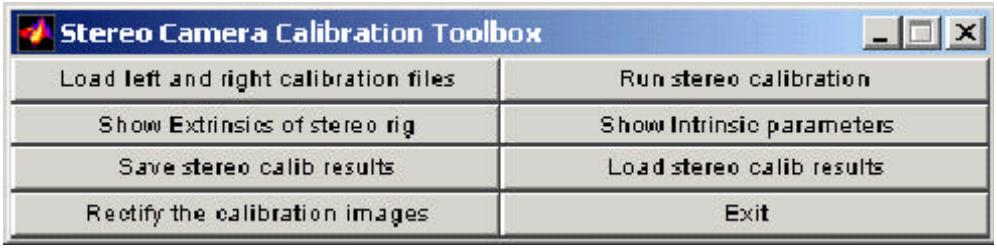


Figura 21 - Stereo Calibration Toolbox di Matlab

Questo ulteriore processo consente di determinare i parametri estrinseci del sistema; infatti prendendo come riferimento quello relativo ad una delle due camere, il software calcola, sulla base dei parametri intrinseci precedentemente trovati, i valori delle matrici ***R*** e ***T*** per passare da un sistema di riferimento solidale con la camera di sinistra, a quello relativo alla camera destra. Queste due matrici verranno poi utilizzate nel processo di ricostruzione tridimensionale.

```
Stereo calibration parameters:


Intrinsic parameters of left camera:

Focal Length:          fc_left = [ 456.20338   456.51157 ] ± [ 2.93775   2.88726 ]
Principal point:       cc_left = [ 164.56674   137.50349 ] ± [ 4.31266   3.08513 ]
Skew:             alpha_c_left = [ 0.00000 ] ± [ 0.00000  ]
Distortion:            kc_left = [ -0.06407   1.19693   0.00954   0.00001  0.00000 ]


Intrinsic parameters of right camera:

Focal Length:          fc_right = [ 459.82036   458.28958 ] ± [ 2.89451   2.75439 ]
Principal point:       cc_right = [ 172.91743   132.65944 ] ± [ 4.62726   2.67951 ]
Skew:             alpha_c_right = [ 0.00000 ] ± [ 0.00000  ]
Distortion:            kc_right = [ -0.01818   0.29101   0.00907   0.01082  0.00000 ]


Extrinsic parameters (position of right camera wrt left camera):

Rotation vector:             om = [ -0.00824   -0.07033  -0.00628 ]
Translation vector:           T = [ -97.70769   0.98845  -12.19053 ]
```

Figura 22 - Parametri intrinseci ed estrinseci derivanti dalla calibrazione

Dalla tabella notiamo le lunghezze focali, i punti principali, i parametri di distorsione, il vettore di rotazione e il vettore di traslazione. La misura di quest'ultimo è stata verificata misurando l'effettiva distanza delle due telecamera, mentre si è notato una differenza delle componenti lungo *z* e *y* causata dalla non perfetta complanarità delle telecamera, che sarà successivamente tenuta in considerazione nel processo di rettificazione.

Terminata la procedura di calibrazione è possibile visualizzare su di un grafico tridimensionale la posizione relativa delle due telecamere e la posizione successivamente assunta dal pattern di calibrazione; in questo modo si può effettuare una rapida valutazione del volume di spazio effettivamente calibrato.

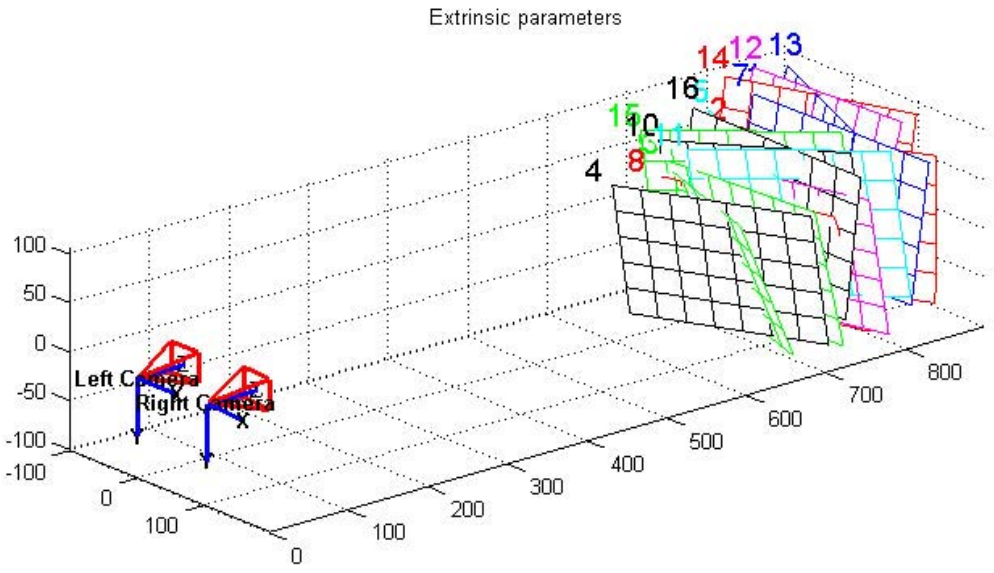


Figura 23 - Ricostruzione tridimensionale delle scene di calibrazione

4.3 Determinazione delle Matrici di Proiezione Prospettica

Ottenuti i parametri intrinseci ed estrinseci della calibrazione stereo vengono calcolati tramite la funzione Matlab *mpp.m* le matrici di proiezione prospettica come composizione delle matrici A, R e T. In particolare per la matrice di proiezione sinistra si considera R come matrice identità e T come vettore di zeri in quanto il sistema di riferimento considerato è attualmente con origine nella telecamera sinistra.

Le matrici calcolate vengono quindi salvate nel file *dati.mat* e i parametri intrinseci ed estrinseci salvati nel file *datitriangolazione.txt*. Per calcolare le nuove matrici di proiezione prospettica e le matrici di trasformazione rettificante viene invocata la funzione *rect.m*. Per una spiegazione più dettagliata si rimanda al paragrafo della rettificazione. Le matrici di trasformazione rettificante vengono salvate nel file *datirettifica.txt* per poi essere prelevati dall'applicazione Java.

Questa è l'ultima operazione effettuata off-line, di seguito verranno esposte tutta la sequenza delle operazioni in real-time.

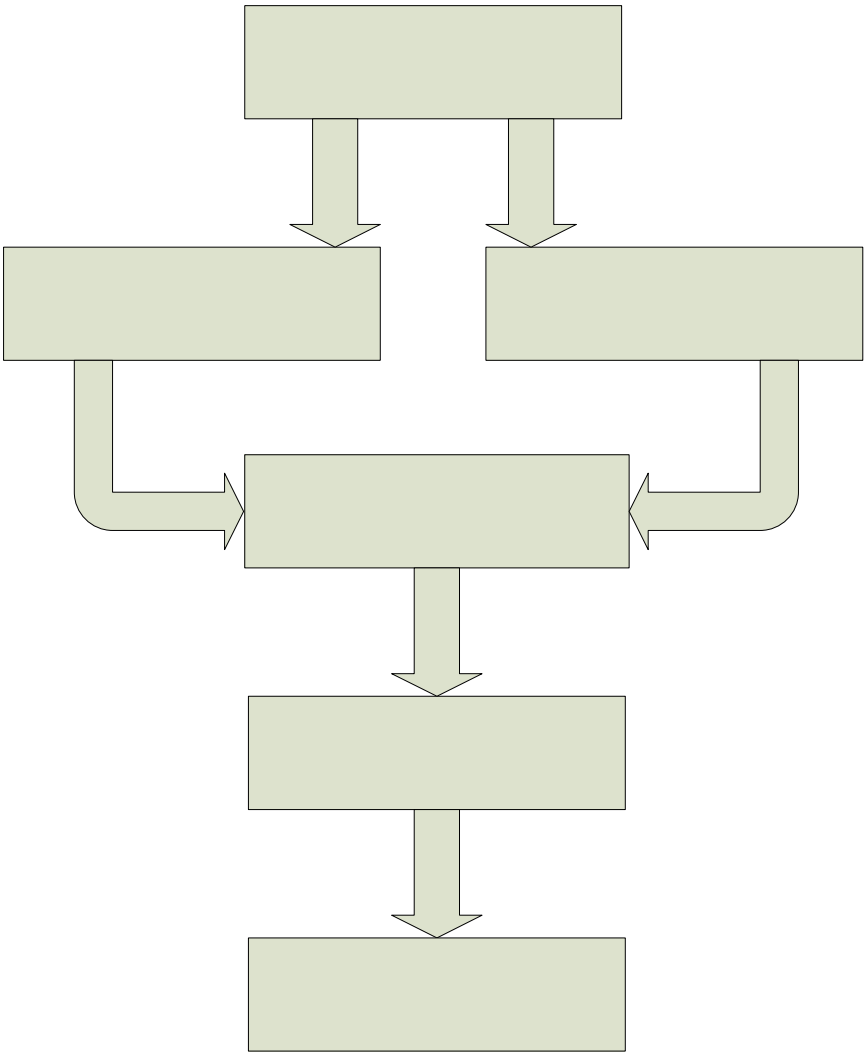


Figura 24 - Processo di calibrazione

4.4 Acquisizione del flusso video e memorizzazione dei frame

La fase iniziale dell'elaborazione in real-time consiste nell'acquisizione di un flusso video dal quale si estraggono le singole immagini, i frame, da elaborare al fine di determinare le informazioni utili alla navigazione del robot e all'individuazione del suo obiettivo. Innanzitutto vengono inizializzate le telecamera e per ognuna di esse vengono caricati i relativi player utili all'acquisizione del flusso.

In una seconda fase viene realizzato un meccanismo di grabbing dei frame che permette di fermare le immagini provenienti dal flusso video.

L’applicazione sviluppata prevede sia la possibilità di visualizzare un’anteprima del flusso video sia la possibilità di estrazione dei frame in maniera manuale.

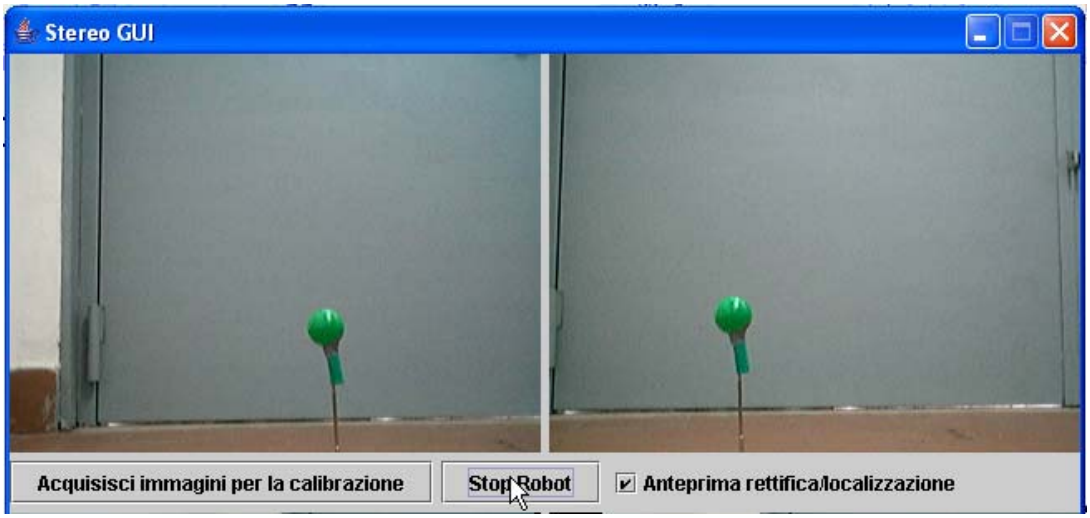


Figura 25 - Finestra di anteprima del flusso video con oggetto da localizzare

Visto che l’elaborazione va eseguita in modalità real-time, i frame provenienti dal flusso video vanno memorizzati temporaneamente in un buffer in maniera tale da ottimizzare l’allocazione della memoria su disco fisso ed anche perché, una volta elaborate, le immagini non contribuiscono più all’apporto di informazioni. I buffer utilizzati per le immagini, costituiti da matrici, hanno una dimensione pari a 320x240x3 ed un formato compatibile al sistema RGB. Successivamente, i frame acquisiti e bufferizzati verranno rettificati per ovviare al problema delle corrispondenze dei punti coniugati, rendendone computazionalmente efficiente la ricerca.

4.5 Rettifica delle immagini

Abbiamo già visto che una delle problematiche della visione stereoscopica consiste nell'accoppiamento tra punti delle due immagini che sono proiezione dello stesso punto nella scena. Il processo di rettificazione consiste nella ricerca delle corrispondenze di tali punti. La ricerca diventa computazionalmente meno onerosa se sfruttiamo un vincolo detto "vincolo epipolare", che permette un'analisi unidimensionale invece che bidimensionale.

Acquisito il flusso video, le immagini stereoscopiche vengono rettificate e possono essere pensate come acquisite da un nuovo sistema di riferimento ottenuto per rototraslazione delle telecamere originarie. Di seguito viene mostrata una coppia di immagini acquisite durante la calibrazione e poi rettificate.

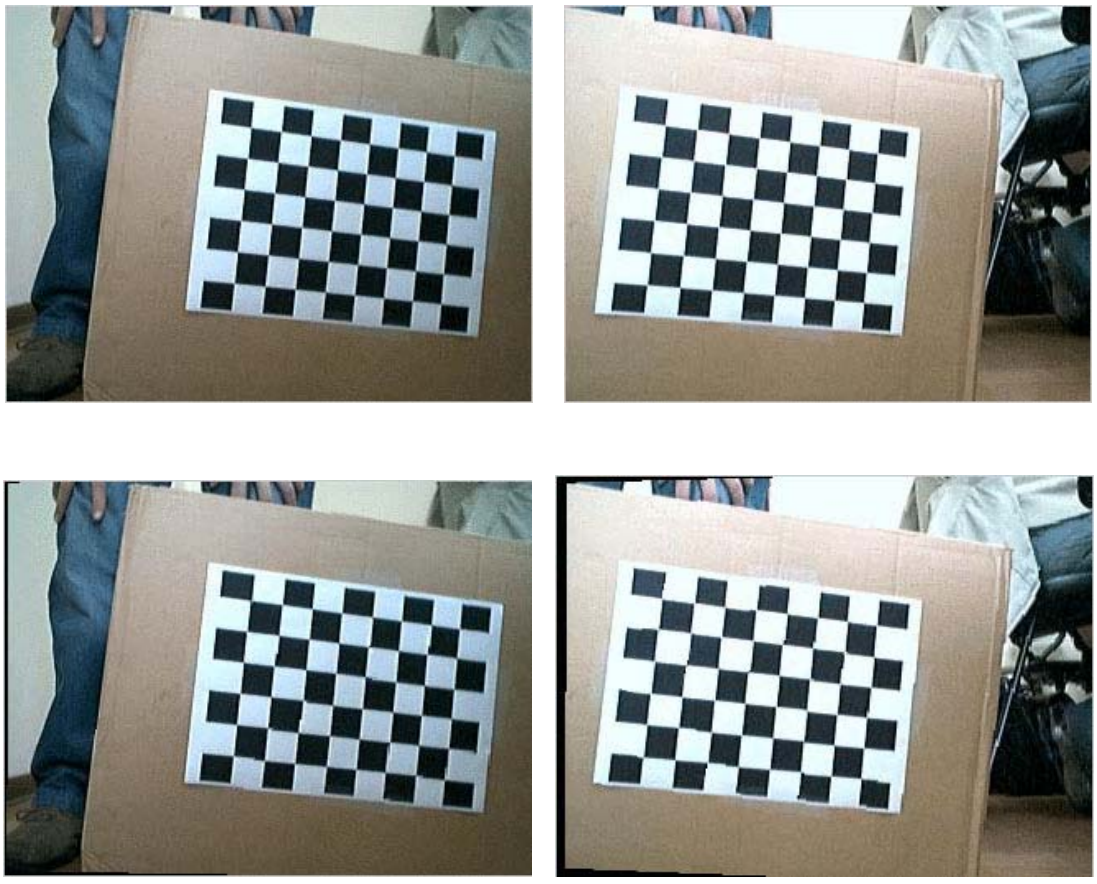


Figura 26 - Immagini prima e dopo la rettificazione

Grazie alla rettificazione le rette epipolari sono tutte orizzontali, parallele e collineari. Effettuata la rettificazione epipolare si procede all'interpolazione bilineare che provvede a colmare i vuoti dovuti ai pixel che presentavano coordinate non intere. La tecnica sopra descritta viene implementata in JAVA mediante l'utilizzo della funzione "InterpolationNearest" presente nelle librerie Java Advanced Imaging.

La funzione che applica la trasformazione ai frame provenienti dal flusso video è *rectify.java*. Tale funzione rettifica i frame acquisiti utilizzando le matrici di rettifica precedentemente caricate (dal file *datirettifica.txt*). In particolare la funzione di rettificazione effettua le seguenti operazioni:

- Le coordinate di ogni pixel vengono tradotte in coordinate omogenee;
- A ciascun punto di coordinate omogenee applica la trasformazione rettificatrice prima calcolata;
- Torna alle coordinate cartesiane e normalizza, applicando a ciascun punto la distorsione radiale e tangenziale ottenendo le coordinate distorte di ogni pixel;
- Applica il metodo dell'interpolazione bilineare con le coordinate distorte;
- A ciascun punto viene associato il valore cromatico RGB.

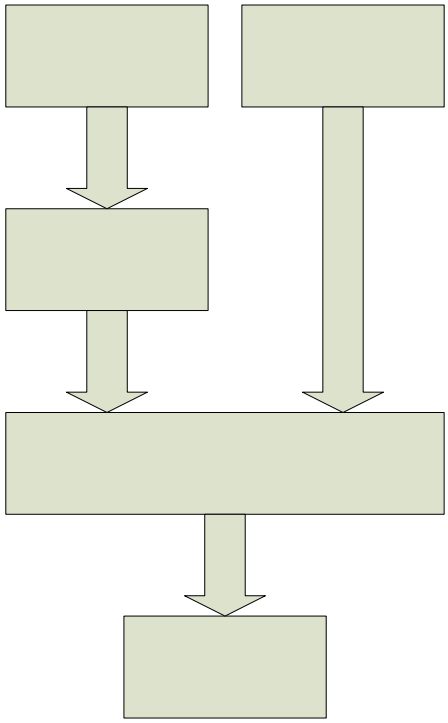


Figura 27 - Schema a blocchi del processo di rettificazione

4.6 Ricostruzione 3-D per triangolazione

Noti i parametri intrinseci ed estrinseci del sistema è possibile effettuare una ricostruzione completa per triangolazione. Tale ricostruzione è infatti immediata, in teoria è sufficiente intersecare i due raggi passanti per i centri ottici e le proiezioni di *P* sulle immagini di sinistra e di destra per determinarne le coordinate 3-D mondo. Nella pratica a causa di approssimazioni i raggi non si intersecano, quindi si cercano i punti con minima distanza da entrambi i raggi.

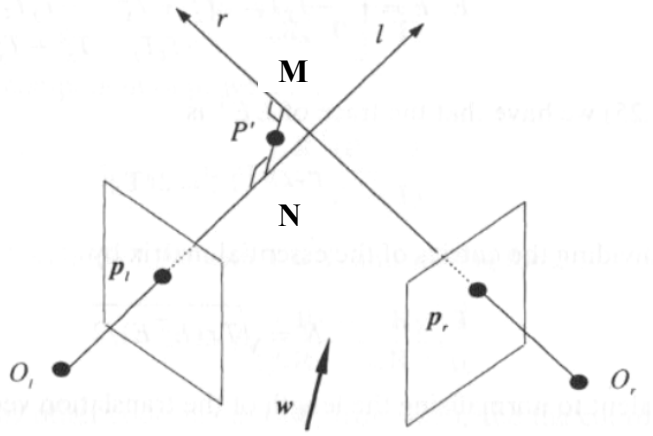


Figura 28 - Triangolazione

Le equazioni parametriche che descrivono i raggi ottici

$$\begin{cases} P' = O_l + \lambda_1 \cdot (An \cdot Rn)^{-1} \cdot p_l \\ P' = O_r + \lambda_2 \cdot (An \cdot Rn)^{-1} \cdot p_r \end{cases}$$

possono essere espresse attraverso i parametri direttori:

$$\begin{cases} (l_1, m_1, n_1) = (An, Rn)^{-1} \cdot p_l \\ (l_2, m_2, n_2) = (An, Rn)^{-1} \cdot p_r \end{cases} \quad \Rightarrow \quad \begin{cases} \begin{bmatrix} \mathrm{x} \\ \mathrm{y} \\ \mathrm{z} \end{bmatrix} = \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} + \lambda_1 \begin{bmatrix} l_1 \\ m_1 \\ n_1 \end{bmatrix} \\ \begin{bmatrix} \mathrm{x} \\ \mathrm{y} \\ \mathrm{z} \end{bmatrix} = \begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix} + \lambda_2 \begin{bmatrix} l_2 \\ m_2 \\ n_2 \end{bmatrix} \end{cases}$$

Con $P' = (x, y, z)$, $O_l = (x_1, y_1, z_1)$, $O_r = (x_2, y_2, z_2)$.

Per determinare P' occorre prima calcolare le coordinate di *M* e di *N*, per poi esprimerlo come punto medio $\overline{MN}$. Derivando parzialmente l'espressione della distanza euclidea si ottiene la seguente soluzione per λ_1 e λ_2.

$$\begin{bmatrix} \lambda_1 \\ \lambda_2 \end{bmatrix} = \begin{bmatrix} \left(l_1^2 + m_1^2 + n_1^2\right) & -\left(l_1 l_2 + m_1 m_2 + n_1 n_2\right)^{-1} \\ \left(l_1 l_2 + m_1 m_2 + n_1 n_2\right)^{-1} & \left(l_2^2 + m_2^2 + n_2^2\right) \end{bmatrix}^{-1} \cdot \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}$$

dove $b_1 = (O_R - O_l)^T \cdot \begin{bmatrix} l_1 \\ m_1 \\ n_1 \end{bmatrix}$ $\qquad b_2 = (O_R - O_l)^T \cdot \begin{bmatrix} l_2 \\ m_2 \\ n_2 \end{bmatrix}$

quindi $\begin{cases} \begin{bmatrix} \mathrm{x_M} \\ \mathrm{y_M} \\ \mathrm{z_M} \end{bmatrix} = \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} + \lambda_1 \begin{bmatrix} l_1 \\ m_1 \\ n_1 \end{bmatrix} \\ \begin{bmatrix} \mathrm{x_N} \\ \mathrm{y_N} \\ \mathrm{z_N} \end{bmatrix} = \begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix} + \lambda_2 \begin{bmatrix} l_2 \\ m_2 \\ n_2 \end{bmatrix} \end{cases}$

E' possibile quindi calcolare le coordinate di $P' = (x_{P'}, y_{P'}, z_{P'})$:

$$x_{P'} = \frac{x_M - x_N}{2} \qquad y_{P'} = \frac{y_M - y_N}{2} \qquad z_{P'} = \frac{z_M - z_N}{2}$$

Tali coordinate si riferiscono ad un sistema di riferimento con origine sulla telecamera sinistra e asse X verso le telecamera destra. È stato cambiato tale sistema di riferimento considerando l'origine al centro delle due telecamere. Tale scelta è stata adottata cosicché tutte le coordinate siano riferite all'asse di simmetria del robot.

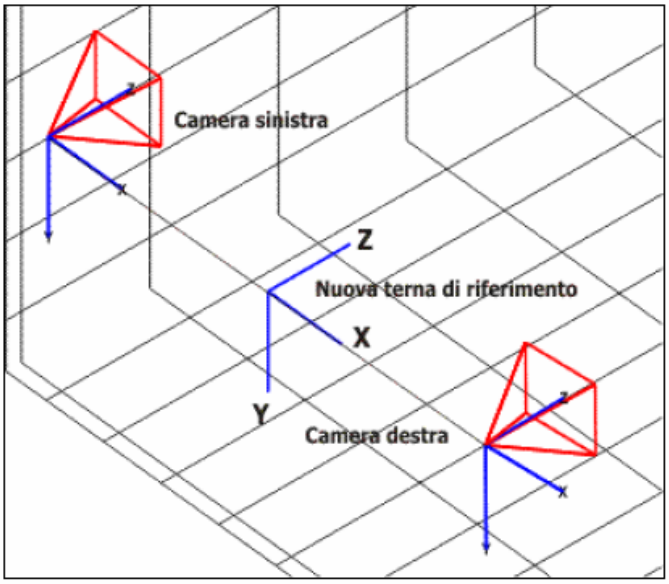


Figura 29 - Nuovo sistema di coordinate riferito al centro del robot

4.7 Localizzazione

Il processo di localizzazione consiste nell'individuare l'oggetto da inseguire. Facendo uso delle immagini rettificate, mediante la triangolazione, viene calcolata la posizione dell'oggetto rispetto al robot, nelle coordinate mondo. Tale processo è stato pensato per essere applicato all'architettura real-time del sistema, quindi si è cercato di sviluppare un algoritmo con ridotti tempi di elaborazione. La migliore soluzione è stata quella di individuare l'oggetto mediante le sue componenti RGB. Come introdotto nella rettificazione la ricerca nell'immagine di destra risulta semplificata in quanto avviene lungo una sola dimensione. Mostriamo di seguito in maniera dettagliata l'algoritmo di localizzazione.

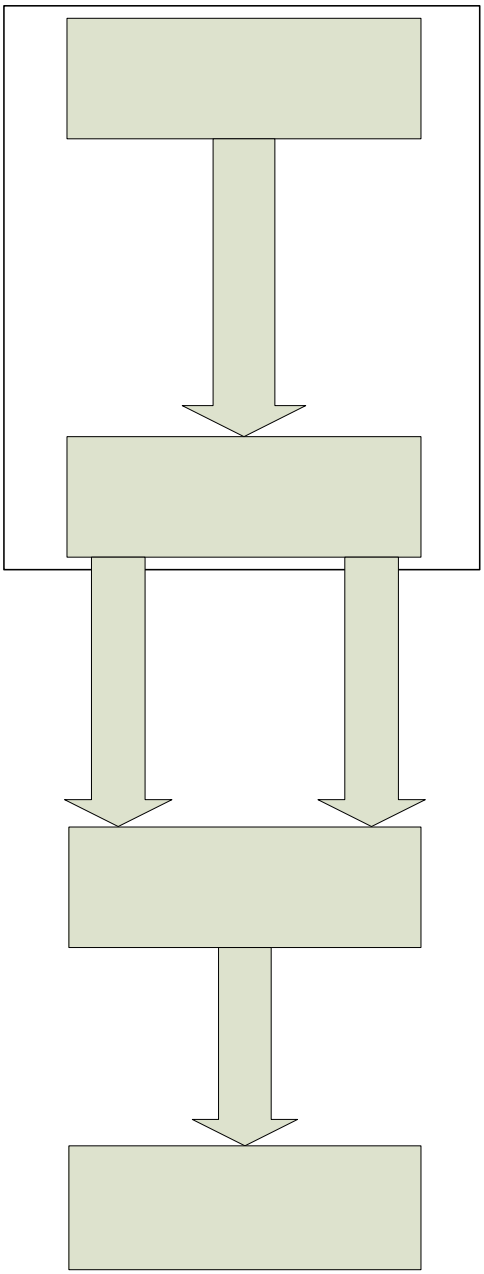


Figura 30 - Schema a blocchi dell'algoritmo di localizzazione

Il processo di localizzazione prevede inizialmente la ricerca dell'oggetto nell'immagine di sinistra mediante le sue componenti RGB. Tale ricerca che prevede la scansione dell'intera immagine rettificata, si arresta appena individuato la terna di valori RGB indicati. Acquisite le coordinate del punto queste vengono memorizzate e viene effettuata la ricerca nell'immagine rettificata di destra. In questa fase la scansione viene effettuata lungo una sola dimensione dell'immagine, in particolare lungo la riga individuata nell'immagine di sinistra. La ricerca in un'unica dimensione nell'immagine di destra è stata possibile grazie alla particolarità delle immagini rettificate, che

hanno i punti coniugati su linee epipolari parallele, orizzontali e collineari. L'applicazione Java implementata permetta la visualizzazione delle immagini rettificate con eventualmente evidenziato l'oggetto localizzato. Inoltre è possibile attraverso la finestra di sistema visualizzare le coordinate mondo dell'oggetto stesso.

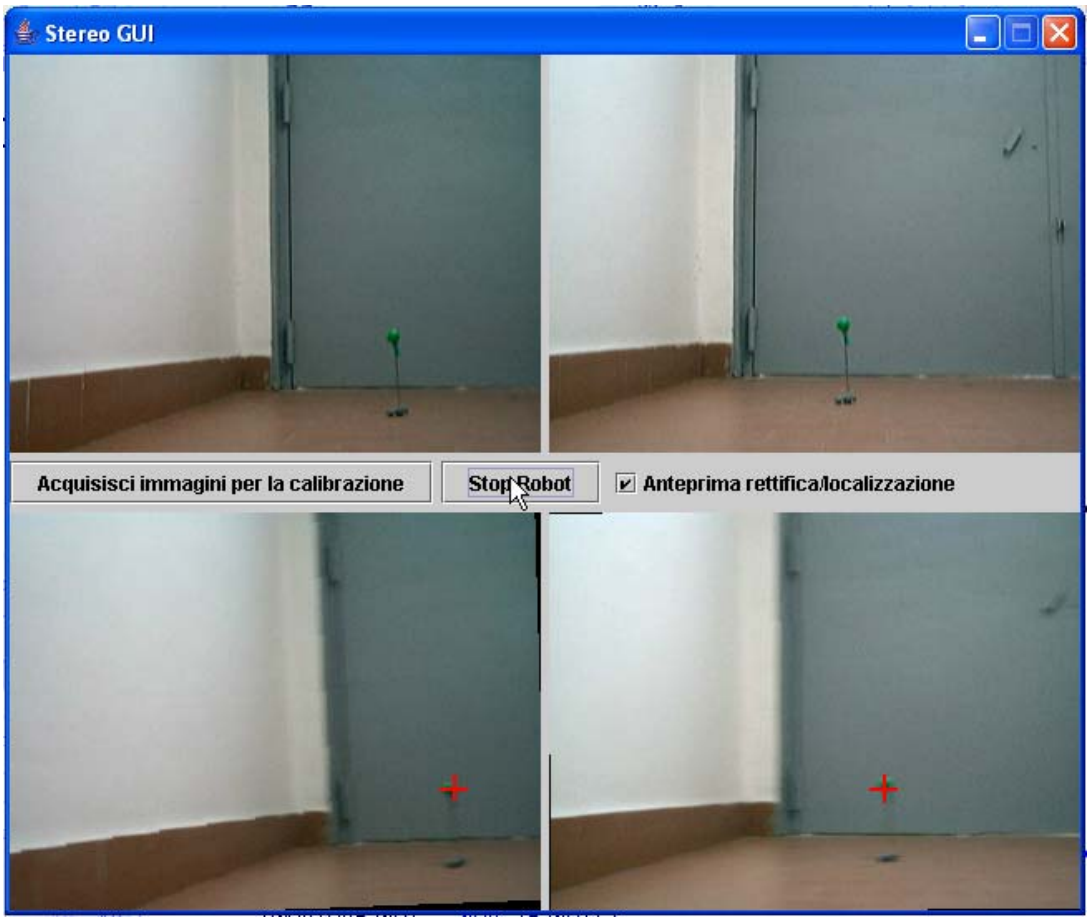


Figura 31 – Anteprima di localizzazione

L'algoritmo sviluppato per la ricerca dell'oggetto prevede che il Robot effettui inizialmente una rotazione su se stesso in senso orario per esplorare l'ambiente circostante. Se la telecamera di sinistra riesce a rilevare l'oggetto il Robot inverte il senso della rotazione in maniera tale da puntare l'oggetto. Se anche la telecamera di destra riesce a rilevare l'oggetto allora viene effettuata la triangolazione, viene restituita la profondità dell'oggetto sulla scena e il Robot si muovera verso l'oggetto. Il moto verso l'oggetto è controllato ad istanti regolari di tempo cioè, la traiettoria da seguire viene ricalcolata

ricorsivamente in maniera tale da riuscire ad avere sempre traccia della posizione dell'oggetto. Il robot si fermerà in prossimità dell'oggetto quando questo si troverà ad una distanza minima precedentemente stabilita. Nel caso in cui il Robot abbia cambiato traiettoria dopo aver individuato l'oggetto nella telecamera di sinistra e se non si ha la conferma dell'individuazione nella telecamera di destra, verrà nuovamente invertito il senso di rotazione in maniera tale da proseguire la ricerca dell'oggetto.

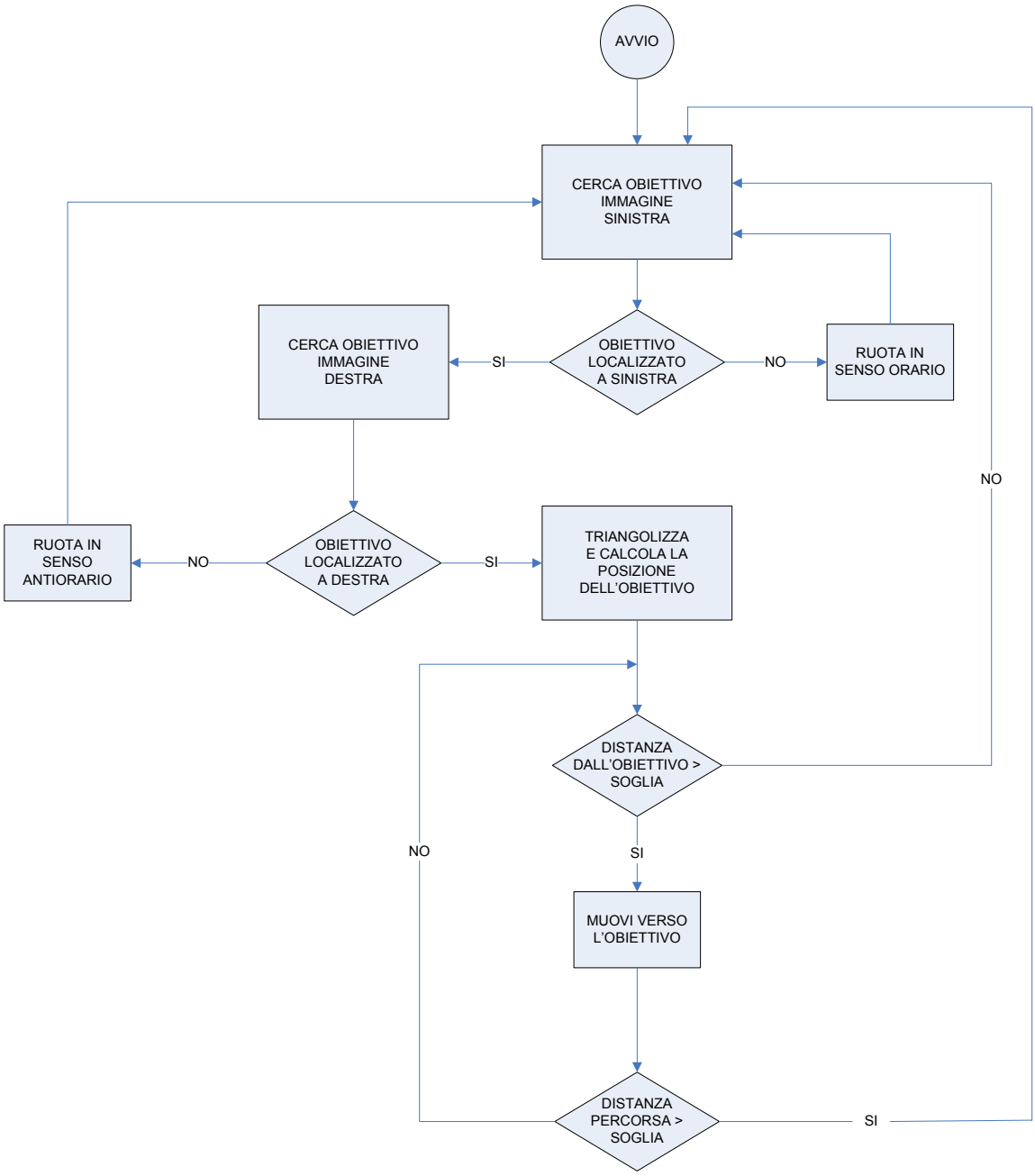


Figura 32 - Diagramma di flusso per la ricerca dell'obiettivo


Stereo GUI
Acquisisci immagini per la calibrazione
Stop Robot
Anteprima rettifica/localizzazione


Stereo GUI
Acquisisci immagini per la calibrazione
Stop Robot
Anteprima rettifica/localizzazione

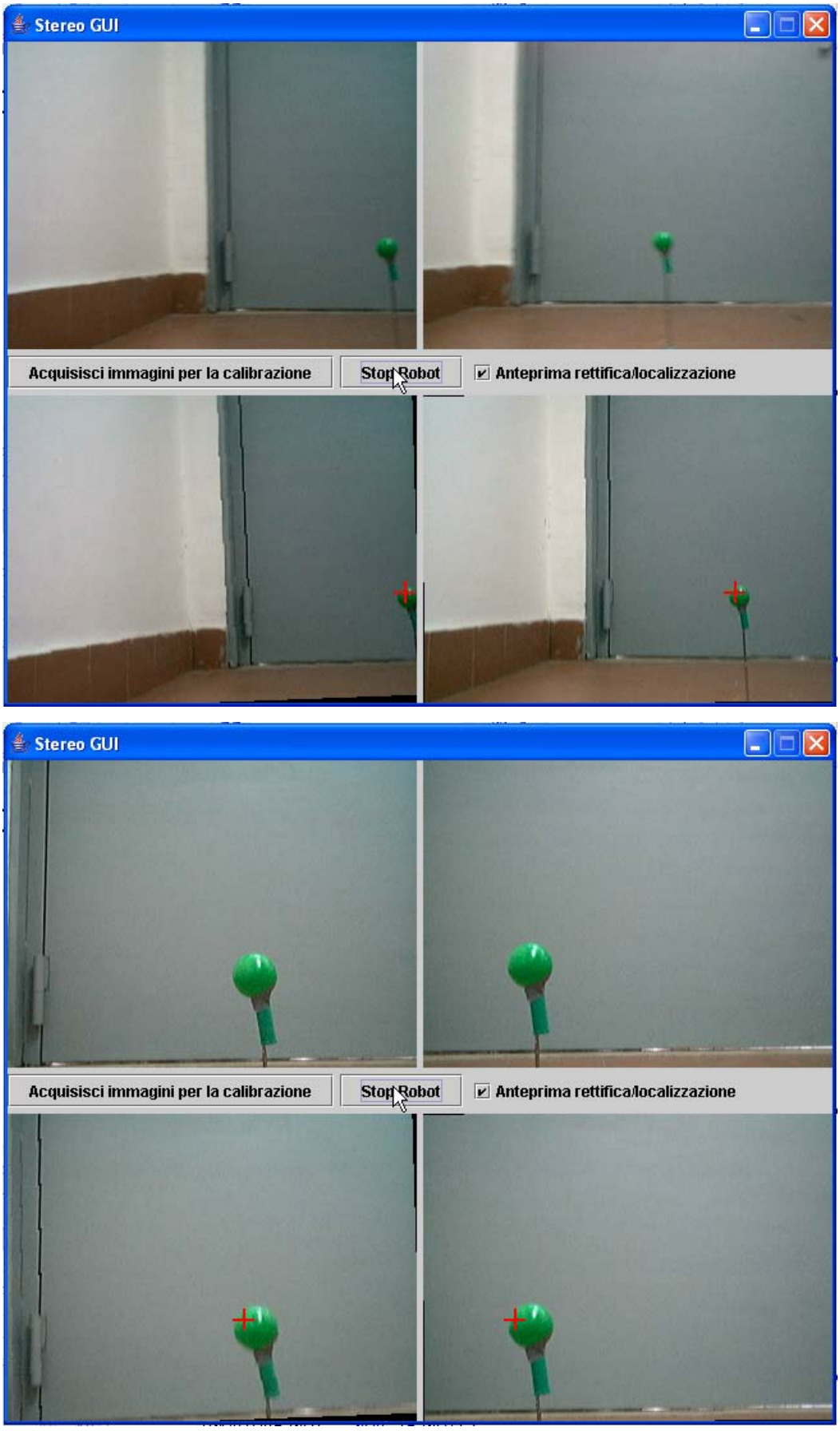


Figura 33 - Frame catturati durante il processo di localizzazione

5. Conclusioni

In questo lavoro è stato progettato e realizzato un sistema hardware/software basato su una piattaforma mobile dotata di visione stereoscopica. Sono stati studiati ed implementati degli algoritmi di visione artificiale per il riconoscimento e l'inseguimento automatico di oggetti, in particolare sono state realizzate ed applicate strategie di controllo per la movimentazione del robot mobile.

6. Sviluppi futuri

Un primo miglioramento del sistema sarebbe quello di sfruttare la struttura di tipo Pan & Tilt delle telecamera, in grado di garantire due gradi di mobilità alla vista del robot. In questo modo l'oggetto potrebbe essere inseguito o cercato girando semplicemente gli "occhi" piuttosto che il corpo del robot. Inoltre si potrebbe migliorare l'algoritmo di localizzazione, che si basa sulle sole componenti di colore, in maniera tale da individuare l'obiettivo anche in funzione della sua forma geometrica mantenendo i tempi di elaborazioni tali da permettere il funzionamento in real-time.

7. Appendice

A Elenco software e librerie utilizzate

Matlab 7

- Matlab Image Processing Toolbox
- Matlab Image Acquisition Toolbox
- Calibration Toolbox
- Stereo Calibration Toolbox

JCreator 3.5.10 Professional

Microsoft Graphedit

Librerie Java

- Java SDK SE 1.4.2.12
- Java Comm Interface
- Java Media Framework 2.1.1e (JMF)
- Java Advanced Imaging JDK 1.1.2.1 (JAI)
- Java 3D 1.3.1

Driver telecamera Logitech QuickCam 8.4.8

B Elenco classi e funzioni implementate

Funzioni Matlab

- ***mpp.m*** Calcola le matrici di proiezione prospettica a partire dei parametri calcolati tramite il Matlab Calibration Toolbox. Salva i parametri intrinseci ed estrinseci nel file *"datitriangolazione.txt"* e salva inoltre gli stessi parametri unitamente alle matrici di proiezione prospettica nel file *"dati.mat"*;

- ***rect.m*** Carica i parametri intrinseci ed estrinseci dal file *"dati.mat"* e calcola le nuove matrici di proiezione prospettica e le matrici di trasformazione rettificanti utilizzando la funzione *rectify* definita al suo interno. Salva i risultati nel file *"datirettifica.txt"*.

Classi Java

- ***GuiMain.java*** Contiene il main e richiama le funzioni per il caricamento delle telecamera, la connessione e l'avvio del robot;

- ***LoadCam.java*** Carica i device video e avvia i player per l'acquisizione del flusso video. Utilizza le librerie JFM;

- ***StartRobot.java*** Avvia il processo acqusizione-rettificazione-localizzazione-triangolazione-movimento;

- ***KoalaInterface.java*** Stabilisce la connessione seriale tra l'unità di controllo e il robot e implementa i metodi per l'interfacciamento.

- ***Rectify.java*** Rettifica i frame acquisiti utilizzando le matrici di rettifica precedentemente calcolate.

- ***Localize.java*** Localizza l'obiettivo sui frame rettificati restituendone le coordinate.

- ***Triangulation.java*** Calcola la posizione mondo dell'obiettivo facendo uso dei parametri intrinseci ed estrinseci precedentemente calcolati in funzione delle coordinate ricevute.

- ***MoveToPoint.java*** Invia i comandi di movimento al Robot in funzione della posizione mondo calcolata. Implementa inoltre il comportamento di rotazione per la ricerca.

Di seguito viene mostrato il diagramma delle classi sopra descritte.

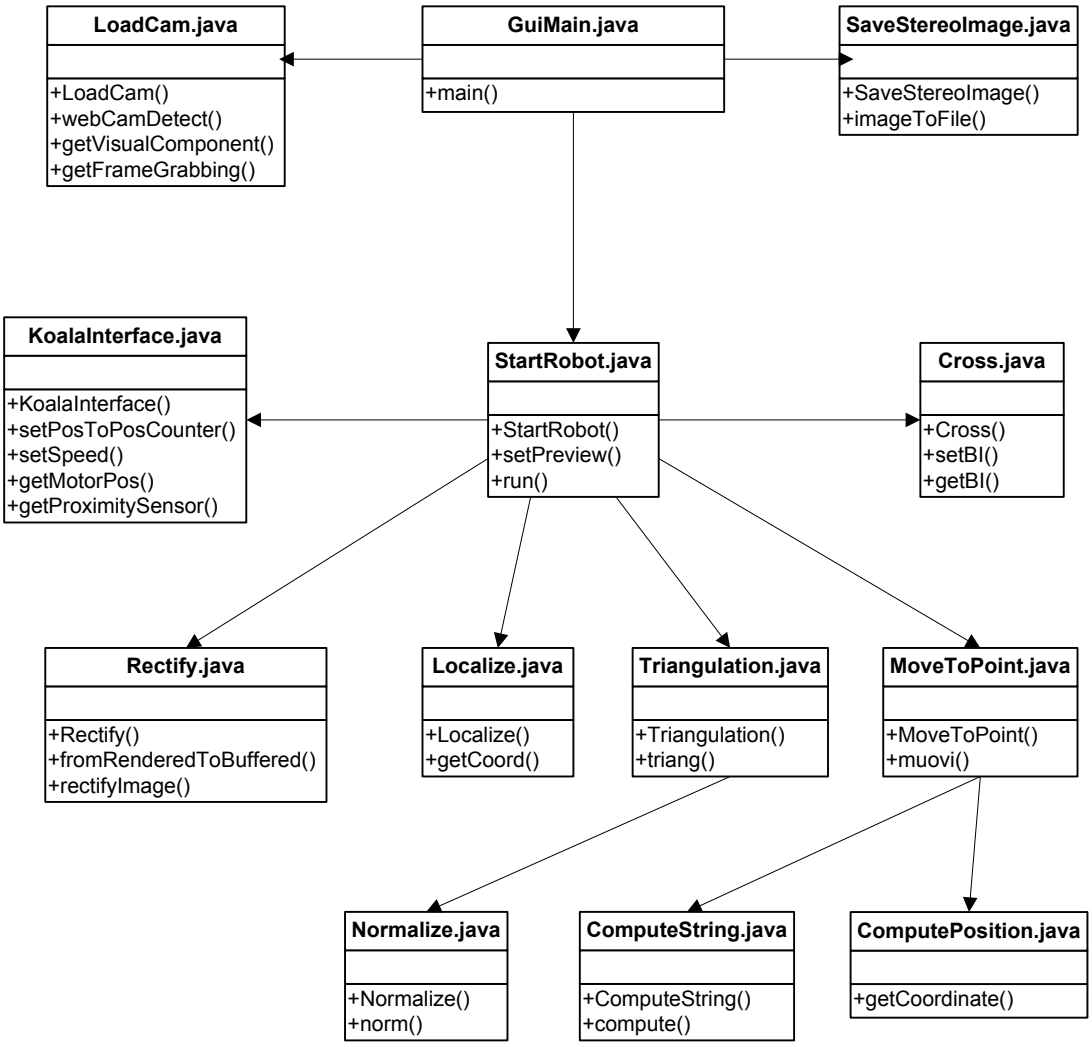


Figura 34 - Diagramma delle classi

C Installazione librerie, configurazione registri e software

Procedura relativa a Windows Xp

1. Installare Java SDK SE 1.4.2.X
2. Installare Java Media Framework 2.1.1.e (JMF)
3. Installare Java Advanced Imaging JDK 1.1.2.1 (JAI)
4. Installare Java 3D 1.3.1
5. Copiare i file relative alla libreria Java Comm API
   - Comm.jar in /jre/lib/ext/
   - Javacomm.properties in /jre/lib/
   - Win32com.dll nella cartella di lavoro
6. Installare i driver delle telecamera
7. Collegare i cavi USB
8. Installare i driver dell'adattatore seriale-usb
9. Avviare JMF Registry e su Capture Device selezionare Detect Capture Device
10. Selezionare una delle due telecamera e fare terminare il processo di ricerca
11. Modifica al registro di Windows:

portarsi su

HKEY_LOCAL_MACHINE/SYSTEM/CURRENT CONTROL SET/ CONTROL/MEDIA RESOURCES/MS VIDEO

Fare una copia della chiave MSVideo.VFWWDM e delle stringhe al suo interno in una nuova chiave MSVideo1.VFWWDM.
Annotare il DevicePath

portarsi su

HKEY_LOCAL_MACHINE/SYSTEM/CURRENT CONTROL SET/ ENUM/USB/vid del device path annotato/

In questo percorso sono presenti le chiavi delle due telecamera. Una contiene il riferimento (#.....#) al DevicePath annotato. Copiare il riferimento della seconda telecamera e modificare il valore del DevicePath sulla chiave MSVideo1.VFWWDM creata in precedenza.

12. Riavviare JMF Registry e rifare il Detect Capture Device (verificare la presenza di due device di cattura video)
13. Compilare tutte le classi java (includendo il classpath di tutte le librerie installate)
14. Avviare la classe GuiMain

Bibliografia

[1] Andrea Fusiello. Visione Computazionale: appunti delle lezioni. http://www.sci.univr.it/~fusiello

[2] Ignazio Infantino. Computer Vision & Robotics. Slides per corso di Robotica A. A. 2001 – Università degli Studi di Palermo.

[3] J.Y. Bouguet, "Camera calibration Toolbox for Matlab", http://www.vision.caltech.edu

[4] Z. Zhang, "Flexible Camera Calibration By Viewing a Plane From Unknown Orientations", Microsoft Research, One Microsoft Way, Redmond,WA 98052-6399, USA, 1999.