

Calibration of Visual and Laser Sensor for Robot Teleguide

Alessandro Mazzurco

Giugno-Settembre 2009

Supervisors:

Prof. Salvatore Livatino

University of Hertfordshire, United Kingdom

Prof. Giovanni Muscato

Università degli Studi di Catania, Italy

Indice

Indice.....	2
Introduzione.....	3
1 SVILUPPI PRECEDENTI ED OBIETTIVO.....	4
1.1 Tesi e Report precedenti su 3MORDUC.....	5
1.2 Caratteristiche 3MORDUC.....	6
1.3 Obiettivo.....	9
2 LETTERATURA E BACKGROUND KNOWLEDGE.....	10
2.1 Accurate Motion Estimation and High-Precision 3D Reconstruction by Sensor Fusion [7].....	10
2.2 Extrinsic Calibration of Camera and Laser Range Finder [11].....	13
2.3 Edge Detection.....	14
2.4 Line Detection.....	15
2.5 Corner Detection.....	15
2.6 Considerazioni.....	16
3 CALIBRAZIONE.....	17
3.1 Modello telecamera.....	18
3.2 Parametri intrinseci ed estrinseci.....	19
3.3 Intrinsic Calibration.....	23
3.4 Extrinsic Calibration.....	33
4 CONCLUSIONI.....	39
APPENDICE A.....	40
BIBLIOGRAFIA.....	44

INTRODUZIONE

Il seguente report costituisce un blocco di un lavoro di gruppo sviluppato nei mesi tra Giugno e Settembre 2009 nella *University of Hertfordshire (UK)* grazie al progetto *Erasmus Student Placement*.

Il lavoro è stato condotto sotto la supervisione del prof. S. Livatino (referente locale in UK) e del prof. G. Muscato (referente italiano dell'Università di Catania).

Il lavoro da me realizzato si basa sulla calibrazione del sistema sensoriale costituito dalla coppia di telecamere e dal laser 2D posti sul 3MORDUC, robot mobile situato nell'Università di Catania e teleguidato dall'Università di Hertfordshire.

Inoltre al fine di rendere ciò utile, attraverso una collaborazione di gruppo, come fine ultimo è stato posto l'obiettivo della *sensor fusion* in modo da permettere all'utente che teleguida il robot una più alta percezione dell'ambiente in cui il robot è localizzato mediante anche l'utilizzo di sistemi di visione stereo.

1 SVILUPPI PRECEDENTI ED OBIETTIVO

Il progetto fino ad oggi portato avanti permette la teleguida del robot 3MORDUC in due diverse modalità, teleguida basata su laser o video, attraverso una architettura di rete client-server. Vi è anche una possibilità di navigazione *offline* in un ambiente sintetico ricostruito da un simulatore. Durante la navigazione lo scanner laser, montato sul robot scansiona l'ambiente, raccogliendo dati che in real time vengono elaborati dal server al fine di ricostruire tramite un opportuno algoritmo la mappa dell'ambiente. Quest'ultima, sotto forma di file *bmp*, viene inviata al client, il quale si occupa di elaborarla in modo da ricostruire in real time un ambiente virtuale conforme all'ambiente reale.

Questa modalità presenta alcune problematiche come ad esempio il fatto che:

- non tutti gli ostacoli sono rilevati mediante il laser fisso ad una data altezza dal terreno;
- un uso corretto dei dati laser si ha solo in ambienti parzialmente o privi di luminosità;
- etc...

Questi ed altri sono state le motivazioni che hanno condotto ad un altro sistema di teleguida basato sul video. Sul robot infatti è stato montato, attraverso un supporto rigido, un sistema di telecamere stereoscopiche. Le due telecamere possono essere poste ad una distanza l'una dall'altra, compresa tra i 5 e i 20 cm, inoltre queste ultime sono sincronizzate ad una frequenza di 8 kHz e connesse sfruttando l'interfaccia IEEE 1394 .

In questa modalità di teleguida il server preleva coppie di immagini provenienti dalle due telecamere ed in seguito in un unico pacchetto le invia al client con un frame rate di 1-3 fs/sec.

A questo punto le immagini provenienti dal server, sul lato client, vengono distribuite sugli appositi canali, ovvero quello destro e quello sinistro, in modo da poter essere successivamente processate e visualizzate, in funzione dello strumento di visualizzazione scelto.

Durante la teleguida video si ha quindi un vantaggio dovuto alla semplicità di guida da parte dell'utente che può rendersi conto in maniera più intuitiva, grazie alle immagini reali, della posizione del robot e della distanza che intercorre tra lo stesso e i vari ostacoli.

Uno svantaggio, non trascurabile, si ha a causa delle immagini che richiedono un'occupazione di banda rilevante. Questo causa un *delay time* nella visualizzazione delle immagini che comporta insieme ai problemi di luminosità un rallentamento della guida da parte dell'utente[1].

Per entrambe le modalità di teleguida sono state utilizzate diverse tecniche di visualizzazione con lo scopo di determinare la migliore visione di guida tra *stereo* e *mono vision*. Dagli studi effettuati si è rilevato che la visione stereo permette una maggiore percezione della profondità riducendo così il numero di collisioni del robot con l'ambiente esterno[2].

1.1 Tesi e Report precedenti su 3MORDUC

Nelle tesi precedenti, svolte da A. Pennisi e C. Arena, sono stati valutati i vantaggi della visualizzazione stereoscopica in entrambe le modalità di teleguida laser e video utilizzando diversi approcci alla visualizzazione stereo e diverse *VR facilities* (Flat wall e 3D Laptop). Tutto questo è stato dimostrato, alla fine di un lavoro di modifica del codice esistente, per mezzo di test effettuati su diversi utenti. Durante il lavoro di modifica, Arena e Pennisi, hanno ottimizzato il codice eliminando diversi bug e funzioni inutilizzate, migliorando le prestazioni attraverso dei comandi da tastiera e per finire separando il funzionamento di teleguida video e laser nell'interfaccia client [1]. La tesi svolta da G. Lanteri, invece si concentra principalmente sulla ricerca dello stato dell'arte per quanto riguarda l'*Augmented Reality* basata sulla fusione di informazioni video e laser allo scopo di teleguidare il robot. Inoltre è stato sviluppato la base di un algoritmo che permette l'allineamento dei features dell'immagine ottenuta da stereocamera con i features geometrici ottenuti dalle informazioni del sensor laser (*synthetic images*).

L'algoritmo proposto da Lanteri è suddiviso in:

- Edge detection, utilizzando la funzione `cvCanny()` di `openCV`;
- Line detection, utilizzando la funzione `cvHoughTransform2()` che in base agli argomenti effettua o una trasformato standard di Hough altrimenti una Progressive Probabilistic Hough Transform, quest'ultima per validi motivi è stata preferita;
- Corner detection, utilizzando del codice scritto ad hoc anziché la funzione `cvCornerHarris()` fornita da `openCV`.

L'obiettivo raggiunto è un primo step del prodotto finale desiderato, infatti vi sono ancora punti da approfondire come ad esempio: le informazioni laser possono essere utilizzate per un filtraggio robusto del risultato dell'immagine processing? Come realizzare questo?

Anche il Report di L. Mazzaglia e F. Catania punta al medesimo obiettivo di Lanteri, ovvero creare una nuova modalità di teleguida del robot. In questo caso si vuole raggiungere lo scopo attraverso la fusione della modalità video e quella sul laser. Il lavoro da loro svolto si è basato su una analisi del codice precedentemente sviluppato e la scrittura di una nuova parte di codice con il quale si è ottenuto la sovrapposizione tra l'ambiente simulato in off-line con delle immagini statiche dell'ambiente reale. Questo lavoro si pone come base per un futuro sviluppo di tecniche di allineamento e matching ed inoltre necessita di una sperimentazione attraverso immagini da videocamera.

1.2 Caratteristiche 3MORDUC

Il 3MO.R.D.U.C., “3rd version of the Mobile Robot DIEES University of Catania”, mostrato in Figura 1.2.1, è un robot mobile su ruote (WMR) con guida differenziale.

Questo robot è stato utilizzato con successo in esperimenti di localizzazione e navigazione.

Il movimento è realizzato mediante 2 motori 40W DC, Maxon F2260, e l'albero motore è collegato con una gear box (rapporto di trasmissione 1/19). Quest'ultimo a sua volta è collegato con due ruote di gomma, mentre una terza ruota castor può liberamente muoversi facilitando l'esecuzione delle curve.



Figura 1.2.1: 3MO.R.D.U.C

La struttura del robot ha tre piani collegati tra loro. Sul ripiano più in basso sono poste due batterie al piombo 12 V / 18 Ah che forniscono l'alimentazione garantendo una autonomia di 30-40 minuti di lavoro continuo. Inoltre è posta una board electronic rack che controlla ogni modulo del robot (movimento, sensori e comunicazione)[3].

Sul robot sono presenti inoltre diversi sensori che monitorano lo spazio di lavoro e lo stato del robot e sono di seguito brevemente descritti:

- una cintura di bumpers (16 switches) sono posti per l'intero perimetro sopra le ruote del robot al fine di rilevare e ridurre il numero di collisioni inviando impulsi nel caso di urto;

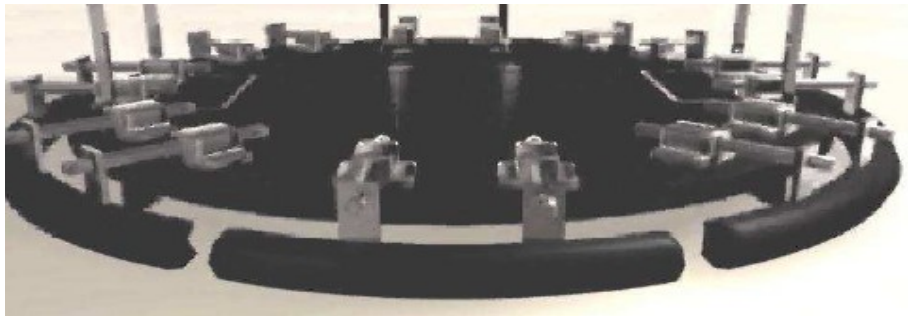


Figura 1.2.2: Bumpers

- i due assi dei motori sono dotati di encoder incrementali o relativi (500 impulsi per giro) che convertono il movimento in una serie di impulsi digitali. Il singolo bit o il set di bits vengono convertiti in una posizione relativa o assoluta.

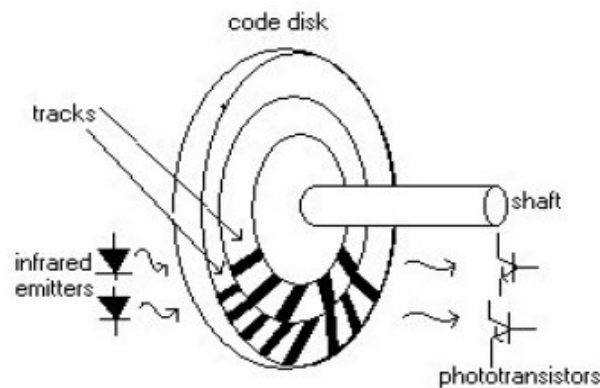


Figura 1.2.3: Encoder relativo.

- 8 sensori sonar sono piazzati intorno al robot per misurare la distanza di eventuali ostacoli basandosi sul tempo di volo di un segnale ultrasonico prodotto grazie alla vibrazione di un sensore piezoelettrico.

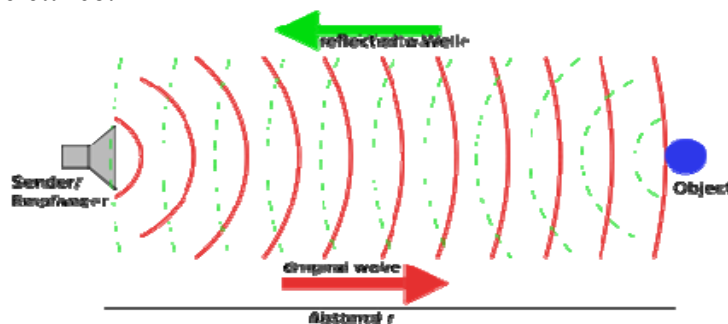


Figura 1.2.4: Sonar sensor.

- uno scanner laser (LMS) effettua delle misurazioni sul tempo di volo di impulsi di luce laser. Un fascio di luce laser viene emesso da una sorgente posta nel sensore e se incontra un oggetto esso viene riflesso. Il fascio riflesso viene registrato dal ricevitore scanner.



Figura 1.2.5: Laser Scanner

Il tempo tra la trasmissione e la ricezione dell'impulso è proporzionale alla distanza tra lo scanner e l'oggetto. Se si indica con L la distanza tra il laser e l'ostacolo e con c la velocità della luce:

$$T_{flight} = \frac{2L}{c} \rightarrow L = c \frac{T_{flight}}{2} \quad (1.2.1)$$

Il fascio laser viene deflesso mediante uno specchio rotante scansionando così una area a forma di ventaglio circostante al sensore. E' possibile configurare diverse risoluzioni angolari ($0.25^\circ/0.5^\circ/1^\circ$) con una diversa massima scansione angolare ($100^\circ/180^\circ$).

Nel 3Morduc si usa una risoluzione angolare di 1° ed una scansione angolare pari a 180° [4].

- una stereocamera STH-MDCS2-VAR-C è posta sul top del robot e connessa al PC mediante l'interfaccia IEEE 1394. Essa è composta da due camere, ciascuna con risoluzione di 1.3 Megapixel. Il sensore CCD di queste camere ha una buona immunità al rumore e sensibilità. Tuttavia è possibile regolare tutti i parametri dell'immagine (exposure gain, frame rate, resolution...). Le due camere sono montate su un supporto rigido che permette di regolarne la distanza fra loro in un range di 5-20 cm. Le immagini provenienti dalle due camere sono sincronizzate ad una frequenza di 8 Khz [5] [6].



Figura 1.2.6: stereocam montata sul 3MO.R.D.U.C.

1.3 Obiettivo

L'obiettivo di questo lavoro è la sensor fusion tra laser e stereocamera. Il risultato che si vuole raggiungere prevede un mapping dei dati laser su ciascuna delle due immagini catturate in modo così da ottenere su tali pixels una terza informazione relativa alla profondità.

Questo permetterebbe una più alta percezione dell'ambiente da parte dell'operatore che teleguida il robot mobile a distanza, infatti non solo avrà informazioni visive ma anche informazioni di distanza che possono essere fornite all'utente mediante una scala di colori o quant'altro.

Per raggiungere tale scopo è necessario calibrare il sistema sensoriale costituito dalla stereocam e dal laser scanner 2D in modo da determinare la trasformazione rigida tra due sistemi di riferimento scelti rispettivamente con l'origine coincidente nel centro dei sensori.

Un esempio di mapping tratto dal *sample* del *Tool Zhang-Pless* è riportato in Figura 1.3.1.

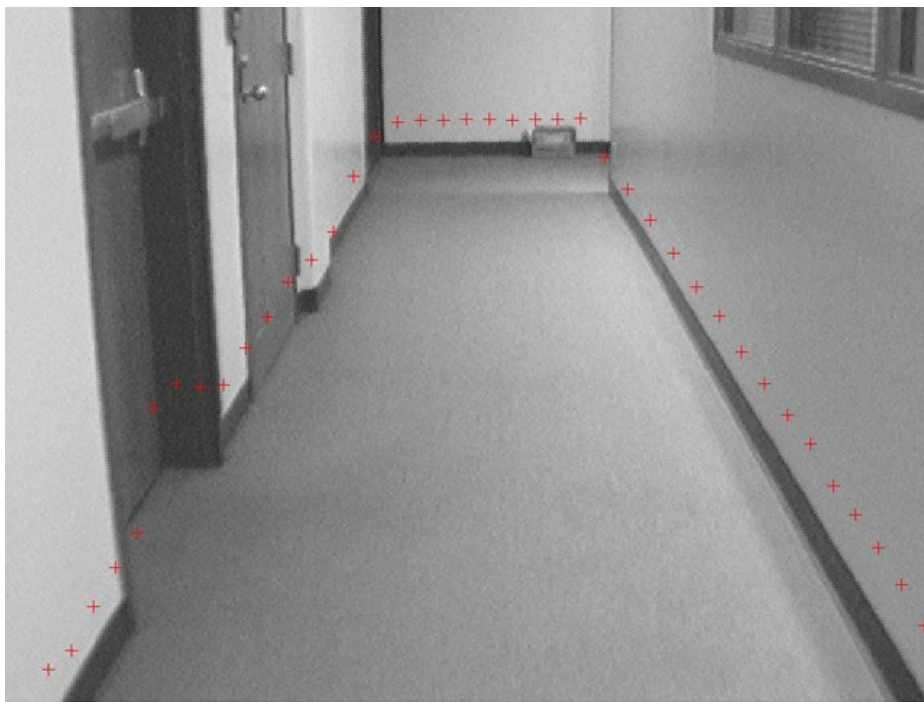


Figura 1.3.1: Mapping laser su immagine, esempio Zhang-Pless

2 LETTERATURA

2.1 Accurate Motion Estimation and High-Precision 3D Reconstruction by Sensor Fusion [7]

Questo articolo ha come obiettivo la Motion estimation e 3D Reconstruction (simultaneamente) mediante l'utilizzo di una CCD camera ed un 2D laser range finder mostrato nella Figura 2.1.1.



Figura 2.1.1: 2D laser range finder con camera

I sensori sono posti come in Figura 2.1.2 per potere inquadrare la stessa scena e dunque puntano alla stessa direzione, inoltre essi hanno una posizione, rispetto ad un sistema di riferimento relativo, fissa.

Per usare immagini e dati laser simultaneamente è necessario trasformare i dati ottenuti dall'ambiente attraverso il sistema sensoriale in un comune sistema di riferimento in modo tale che la posizione relativa tra i due sistemi può essere valutata.

Un metodo per la calibrazione estrinseca tra laser 2D e camera è proposto nel prossimo paragrafo. Prima naturalmente viene effettuata la calibrazione della camera utilizzando un pattern plane con cerchi concentrici [8].

L'idea base per la motion estimation è quella di *3D to 2D matching* sfruttando le informazioni dei punti laser che a differenza di quelli ottenuti solamente da una stereocam hanno un piccolissimo rumore indipendente dalla misura effettuata. Per trovare nell'immagine i dati scansionati si fa una proiezione dei dati laser correnti nell'immagine corrente e poi si trovano i corrispondenti punti laser nell'immagine successiva. Per il tracking viene utilizzato il KLT tracker [9].

Una volta stimato il movimento, i dati scansionati L_i dell' i -esimo frame vengono trasformati nei dati l_i del sistema di riferimento del primo frame. Questo è ottenuto utilizzando il movimento stimato e le relative matrici di rotazione e traslazione di due frames successivi, in questo modo la texture mapping è semplice da realizzare in quanto tutti i dati sono proiettati nella medesima immagine.

I feature points dell'immagine possono essere aggiunti all'immagine 3D e se il matching è corretto la triangolazione genera un più ampio range di informazione di punti 3D rispetto ai dati laser. Tali punti 3D generati dai feature points hanno una grande incertezza rispetto ai 3D laser points lungo la direzione di visualizzazione, inoltre ridurre tale incertezza in molte immagini è impossibile e quindi

nel caso si chieda minore incertezza conviene usare solo dati laser per la ricostruzione 3D. Inoltre in questo articolo è stato elaborato, in particolare, un algoritmo con lo scopo di risolvere un caso degenerativo che si può presentare per la soluzione iniziale ottenuta attraverso il *3-point algorithm* [10], ovvero quando il laser 2D scansiona un piano e si ottiene sull'immagine una linea di dati laser falsi.

Gli esperimenti sono stati fatti sia in ambiente indoor che outdoor.

In outdoor è stato usato il seguente veicolo in cui è stato piazzato lateralmente il sistema di sensori discusso.



Figura 2.1.2: veicolo con sistema laser montato

I sensori sono ruotati di 90° dalla scansione verticale ed un esempio di cattura dati è mostrato in Figura 2.1.3.

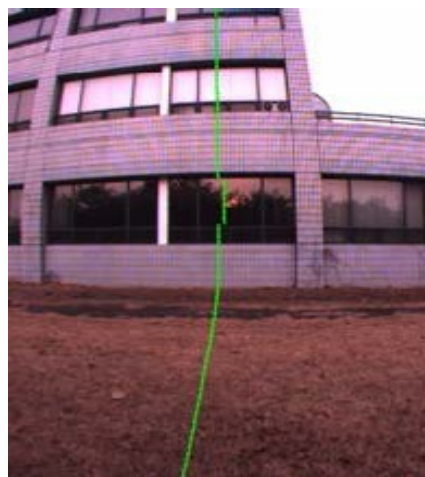


Figura 2.1.3: proiezione dei punti laser sull'immagine da camera (outdoor)

Un esempio dei risultati di motion estimation e ricostruzione 3D sono riportati nelle Figure 2.1.4 e 2.1.5.

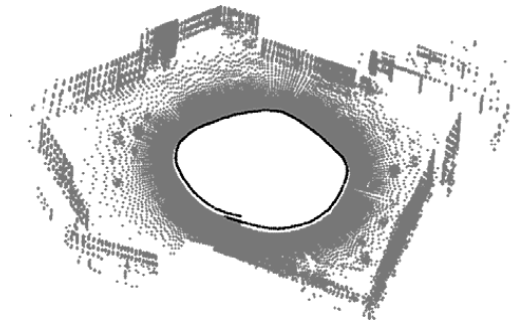


Figura 2.1.4: Mappa Laser, in nero il movimento del veicolo

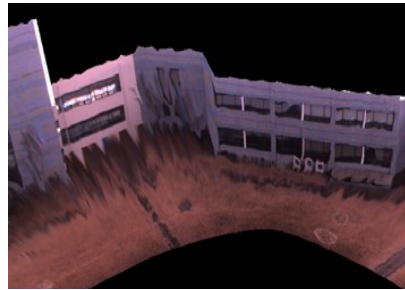


Figura 2.1.5: Texture Mapping

Inoltre nell'articolo sono stati fatti alcuni test con lo scopo di decrementare i tempi di processamento della soluzione iniziale, ottimizzazione e matching.

2.2 Extrinsic Calibration of Camera and Laser Range Finder [11]

Negli ultimi anni i 2D laser sono stati comunemente installati nei robot mobile per vari tasks di navigazione. Essi forniscono in real time un accurato range di misure in un ampio campo angolare ed ad una fissata altezza dal piano di terra. Questi inoltre sono fondamentali per lo sviluppo di sistemi sensoriali che si basano su *data fusion*, infatti insieme a sistemi di visione come una mono o stereo-cam è possibile ottenere una più alta percezione dell'ambiente da parte di un utente che teleguida il robot in cui essi sono installati.

Bisogna inoltre tenere conto che molti sensori hanno un tempo di acquisizione dei dati notevolmente lungo e quindi ciò vuol dire che tali sistemi sensoriali possono operare in modo ottimale in ambienti statici o quasi statici utilizzando così dati acquisiti in un istante precedente a quello corrente.

Inoltre in molti casi esiste una oggettiva limitazione alla larghezza di banda utilizzabile per il collegamento fra il robot e l'operatore. Vi è dunque la necessità assoluta di utilizzare tecniche di pre-acquisizione dei dati e display predittivi.

In questo articolo viene approfondita la tecnica di calibrazione estrinseca tra i 2 sensori. Tale tecnica permette una volta ottenuta la regola di trasformazione tra i sistemi di riferimento di un sensore laser ed una camera, posti entrambi per esempio sul medesimo robot, di stimare la profondità utile per effettuare una texture 3D model.

Vengono considerati noti i parametri intrinseci ed estrinseci della stereocam e quindi è nota la seguente relazione:

$$P_c \approx k(R P_w + t) \quad (2.2.1)$$

dove con k si indicano i parametri intrinseci della camera e con R e t la matrice di rotazione e traslazione della camera rispetto al *sistema mondo* e quindi i 6 parametri estrinseci dovuti ai 6 gradi di libertà. Supponiamo inoltre che la distorsione della telecamera sia stata eliminata dall'immagine. Un sistema di coordinate laser è definito con l'origine coincidente nel dispositivo laser che dunque scansiona il piano $Y=0$. Supposto che P_c sia un punto del sistema di coordinate della camera, esso è localizzato in un punto P_L del sistema di coordinate del laser secondo la seguente relazione:

$$P_L = \Phi P_c + \Delta \quad (2.2.2)$$

dove Φ è una 3x3 matrice ortonormale rappresentante l'orientamento relativo della telecamera rispetto al laser e Δ è un vettore di 3 elementi che congiunge le origini dei 2 sistemi di riferimento, laser e camera.

Lo scopo è proprio quello di determinare posizione ed orientamento della camera rispetto al laser e quindi Φ e Δ .

La calibrazione estrinseca laser-camera si basa principalmente su una soluzione iniziale, una ottimizzazione non lineare ed una globale ottimizzazione utile anche per i parametri intrinseci della camera.

L'algoritmo è così strutturato:

- costruzione di una checkboard e piazzamento davanti al sistema di sensori con diversi orientamenti (entrambi i sensori rilevano il pattern noto come in Figura 2.2.1);
- per ogni posizionamento della checkboard vengono estratti i laser points nelle letture laser e rilevati i punti della griglia della checkboard sull'immagine;
- stima dei parametri Φ e Δ utili per la soluzione iniziale ;
- miglioramento dei suddetti parametri attraverso la minimizzazione non lineare ;
- miglioramento di tutti parametri attraverso la globale ottimizzazione.

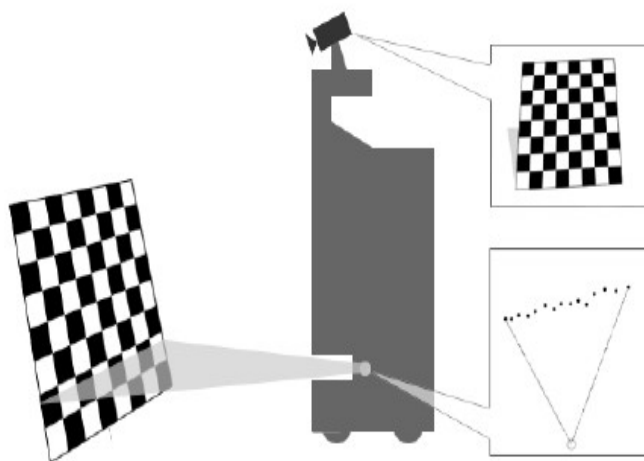


Figura 2.2.1: acquisizione dati per la calibrazione

2.3 Edge Detection

Il contorno di un oggetto rappresenta la separazione tra l'oggetto e lo sfondo o tra l'oggetto ed altri oggetti, per cui la sua estrazione è molto spesso il primo passo verso l'individuazione di esso stesso. Un edge si presenta in una immagine come il confine tra due regioni caratterizzate da proprietà di livello di grigio in qualche modo distinguibili. Esistono diverse tecniche di edge detection le quali principalmente si basano sull'applicazione di un operatore locale di derivata. Le derivate di una funzione digitale possono essere definite in termini delle differenze tra i valori assunti dalla funzione in punti vicini. Qualunque sia il modo con cui si utilizzano queste differenze devono essere soddisfatti i seguenti requisiti.

La derivata prima deve essere:

- nulla nelle zone piatte;
- non nulla in presenza di una discontinuità;
- non nulla lungo una rampa.

La derivata seconda deve essere:

- nulla nelle zone piatte;
- non nulla all'inizio e alla fine di una discontinuità;
- nulla lungo una rampa di pendenza costante.

In generale si è dimostrato che:

- le derivate prime danno luogo a contorni più spessi e hanno una risposta più forte agli edge ripidi;
- le derivate seconde hanno una risposta più forte ai dettagli fini e producono una risposta doppia agli edge.

Dunque il valore della derivata prima può essere utilizzato per determinare la presenza di contorni e gli *zero crossing* della derivata seconda possono consentirne la precisa localizzazione. Il segno della derivata seconda permette di stabilire l'appartenenza di un pixel al versante scuro o al versante chiaro di un contorno. Tra i più robusti edge detector il principale è sicuramente Canny. La qualità dei risultati ottenibili con il metodo Canny si giustifica con il fatto che il metodo utilizza due soglie (hysteresis thresholding) una per l'individuazione degli edge più netti e l'altra per l'individuazione degli edge più deboli. Questi ultimi sono però presi in considerazione solo se risultano connessi ad

edge netti. In openCV è presente tale funzione `cvCanny(in, out, lowthresh, hightthresh, aperture)`.

2.4 Line Detection

Un modo per determinare le linee, che segue l'edge detection, è la trasformata di Hough, ossia un sistema di voting. L'idea base è quella di cercare picchi nella convoluzione tra l'immagine e un insieme di maschere che modellano la struttura locale di rette. Se una maschera M in un punto p ritorna una risposta superiore ad una soglia, diciamo che in quel punto passa un segmento di retta la cui orientazione e spessore è modellato da M . In generale, dati n punti in un piano, stabilire quali di questi punti sono collineari è un problema computazionalmente oneroso. Poiché le linee formate da n punti sono all'incirca n^2 , si tratta di effettuare per ciascun punto altrettanti confronti di distanza quindi in totale circa n^3 confronti.

Un approccio alternativo è proprio la trasformata di Hough che rileva gli estremi di una retta nello spazio dei parametri. Una retta $y=mx+n$ è identificata dalla coppia di parametri (m,n) . Quindi nel caso delle rette lo spazio dei parametri è un piano e in tale piano la retta è rappresentata da un punto. Al contrario un punto nello spazio di partenza (x,y) rappresenta una retta $n=x(-m)+y$ nello spazio dei parametri. Ogni punto di questa linea corrisponde ad una linea nello spazio di partenza che passa per il punto (x,y) . Quindi due punti A e B appartenenti alla stessa retta r , corrispondono nello spazio dei parametri a due rette la cui intersezione ci fornisce i parametri (m,n) della retta r . Dunque una retta nello spazio di partenza definita da N punti P_1, \dots, P_n viene identificata come intersezione nello spazio dei parametri di N rette ognuna corrispondente ad un P_i . I punti di partenza possono essere proprio punti edge e suddividendo lo spazio dei parametri in tante celle e utilizzando il meccanismo di voting è possibile trovare un picco che rappresenta il punto di intersezione e dunque la retta nello spazio di partenza. In openCV esiste una funzione nota con il nome di `HoughTransform2()` che utilizza l'approccio standard appena descritto oppure un altro approccio noto con il nome di Progressive Probabilistic Hough Transform che prende solo alcuni punti dello spazio di partenza e restituisce solo l'estremità della linea riducendo così i tempi di computazione ed eventuali errori sullo spessore delle linee.

2.5 Corner Detection

I corner sono dei punti caratteristici che per loro natura si prestano bene ad essere utilizzati come feature points. L'algoritmo più importante per effettuare corner detection è sicuramente quello dovuto ad Harris e Stephens che calcolano la funzione SSD (sum of squared differences) direttamente come differenza tra porzioni di immagine vicine. Inoltre invece di calcolare la funzione intensità nelle varie direzioni viene considerato uno sviluppo in serie di Taylor nell'intorno del centro della porzione di immagine. Tale algoritmo è invariante per rotazioni dell'immagine e restituisce un certo valore R che permette di stabilire la presenza o meno di corner. In openCV esiste la funzione `cvCornerHarris(const CvArr* image, CvArr* harris_response, int block_size, int aperture_size=3, double k=0.04)`.

2.6 Considerazioni

Entrambe le pubblicazioni, dei primi due paragrafi, risultano essere molto utili nell'ambito della 3D Reconstruction in quanto potrebbero permettere, una volta effettuata la calibrazione del sistema sensoriale laser-camera, di determinare informazioni a partire dai laser points.

Una volta inseriti i dati laser nell'immagine corrispondente mediante le trasformazioni note tra i due sistemi di riferimento, è possibile ottenere una informazione più completa ed accurata dell'ambiente in cui è situato il robot con tale sistema sensoriale installato.

Ciò permette di ridurre offset spaziali tra i dati provenienti rispettivamente da camera e laser ed inoltre permette di trascurare i problemi dati dallo *spot spacing* descritto nei datasheet del laser.

Tuttavia esistono altri articoli in merito alla calibrazione [12] che suggeriscono metodi più facilmente realizzabili. Nonostante ciò non sono stati presi in considerazione per questo lavoro poiché ognuno di essi presentava diverse limitazioni come limitazioni hardware (utilizzo di laser 3D), assenza di un algoritmo di ottimizzazione e mancanza di chiari risultati sperimentali che possano dimostrare la robustezza del metodo utilizzato.

3 CALIBRAZIONE

La calibrazione permette di calcolare i parametri che governano la proiezione operata da una telecamera sui punti dello spazio; cioè permette che ogni punto della scena possa avere il suo corrispondente all'interno dell'immagine. I *punti-scena* ed i *punti-immagine* sono legati tra loro da una "trasformazione" che viene operata dalla telecamera. Il processo di calibrazione consiste nell'identificare questa "trasformazione", ovvero ricavare i parametri che governano il modo in cui un *punto-scena* corrisponde ad un *punto-immagine*. Supponiamo di avere una telecamera che inquadra una scena; si definisce in essa un sistema di riferimento cartesiano tridimensionale xyz con origine a piacere. In tal modo ogni punto della scena risulta essere individuato da una terna cartesiana $(x; y; z)$ ed allo stesso modo, fissato un sistema di riferimento bidimensionale uv , ogni punto dell'immagine può essere individuato da una coppia $(u;v)$ che è dovuta alle immagini bidimensionali prodotte dalla telecamera. In conclusione, sia P un qualunque punto all'interno della scena e P' il suo punto corrispondente nell'immagine prodotta dalla telecamera, il mapping dei punti dello spazio ai punti dell'immagine è mostrato nella Figura 3.1.

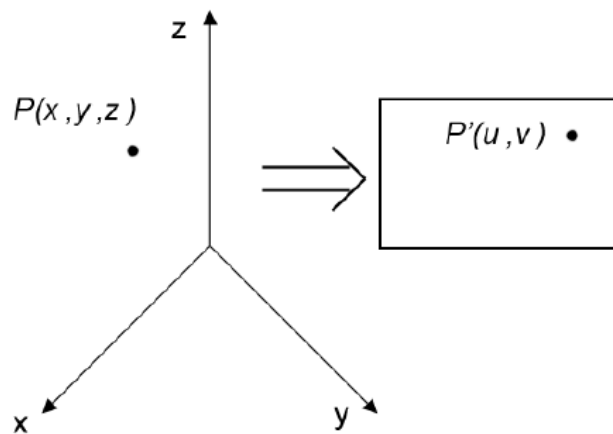


Figura 3.1: Corrispondenza tra punto-scena(P) e punto immagine(P')

E' possibile effettuare anche una trasformazione inversa, che permette, partendo dai punti di un'immagine, di ricavare informazioni sui punti corrispondenti nello spazio. Un problema che si pone in questo tipo di trasformazione è la perdita di informazioni, infatti, il passaggio da un punto dello spazio tridimensionale ad un punto dell'immagine bidimensionale prevede la perdita di una coordinata che tiene conto della profondità. Quindi per i suddetti motivi si ha nel passaggio inverso il problema di recuperare l'informazione relativa alla coordinata persa. Questo problema viene superato da una tecnica nota come *stereopsi* che consiste nel recupero della terza coordinata, utilizzando due o più viste della stessa scena da diverse angolazioni. Tale processo è conosciuto con il nome di *ricostruzione 3D*.

3.1 Modello telecamera

Nel processo di calibrazione, alla telecamera viene sostituito il suo modello geometrico; quindi prima di descrivere come operativamente viene eseguita la calibrazione è utile dare delle delucidazioni su tale modello. Esiste un modello semplice che riesce ad approssimare la realtà: il modello *pinhole* o prospettico. Tale modello risulta essere costituito da due piani uno anteriore, che presenta un foro di dimensioni infinitesime, e uno posteriore. Attraverso il foro passano i raggi luminosi formando nel piano posteriore un'immagine ribaltata dell'oggetto inquadrato. Dal punto di vista geometrico il modello è costituito da un *piano di retina R* (o piano immagine), nel quale si forma l'immagine attraverso una proiezione prospettica, e da un *piano focale F*, in cui non tutti i punti hanno un'immagine nel piano di retina *R*.

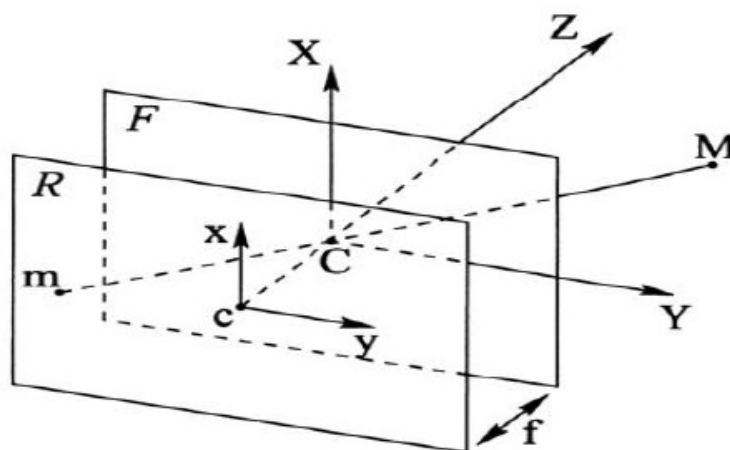


Figura 3.1.1: Modello geometrico della telecamera

Guardando la Figura 3.1.1 possiamo definire:

- f (*distanza focale*) la distanza tra i piani *R* ed *F*;
- il punto *C*, detto *centro ottico* o *fuoco*, che è posto a distanza f dal sistema ottico ed è usato per formare l'immagine *m*, di coordinate (x, y) nel piano *R*, del punto tridimensionale *M* di coordinate (X_c, Y_c, Z_c) , come intersezione di tale piano con la retta passante per *C* ed *M* (*raggio ottico*);
- *asse ottico* la retta passante per il centro ottico *C* e perpendicolare al *piano di retina R* nel quale individua il punto *c* (*punto principale*).

La relazione che intercorre tra le coordinate immagine di *m* nel piano *R* e le coordinate tridimensionali del punto *M* è data da

$$\frac{f}{Z_c} = \frac{-x}{Z_c} = \frac{-y}{Y_c} \quad (3.1.1)$$

$$\begin{cases} x = -f \frac{X_C}{Z_C} \\ y = -f \frac{Y_C}{Z_C} \end{cases} \quad (3.1.2)$$

Dove (X_c, Y_c, Z_c) sono le coordinate del punto M e (x, y) sono le coordinate dell'immagine m [9].

3.2 Parametri intrinseci ed estrinseci

Per la calibrazione della telecamera occorre definire tre diversi sistemi di riferimento:

- il sistema di riferimento 3D detto anche *sistema mondo*;
- il sistema di riferimento standard 3D della telecamera, centrato in C ;
- il sistema di riferimento 2D per l'immagine.

Dato un qualunque punto P della scena, esso potrà essere individuato in ognuno di questi tre diversi sistemi; in particolare si indicherà con:

$X = (X; Y; Z)$ coordinate del punto P nel sistema mondo;

$X_c = (X_c, Y_c, Z_c)$, coordinate del punto P nel sistema della camera;

$x = (x; y)$ coordinate del punto P nel sistema dell'immagine;

$w = (u; v)$ coordinate del punto P nel sistema dell'immagine digitale.

Con l'ultimo sistema si tiene conto della discretizzazione fatta durante la conversione dell'immagine da analogica a digitale.

Esaminiamo la problematica relativa alla coincidenza del sistema mondo con quello standard; questi due sistemi sono collegati tra loro da una trasformazione isometrica composta da una traslazione t e da una rotazione R .

In altre parole esprimendo queste trasformazioni in forma matriciale si ha che il sistema mondo e il sistema standard sono legati tra loro:

$$\begin{bmatrix} X_C \\ Y_C \\ Z_C \\ 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (3.2.1)$$

Come si può notare le coordinate sono espresse in forma omogenea per fare in modo che le operazioni risultino lineari. La precedente equazione si può scrivere in forma abbreviata come:

$$\hat{X}_C = P_R \hat{X} \quad (3.2.2)$$

Si è visto come il sistema di riferimento della telecamera è messo in relazione con quello dell'immagine attraverso la proiezione prospettica:

$$\begin{aligned} x &= -f \frac{X_C}{Z_C} \\ y &= -f \frac{Y_C}{Z_C} \end{aligned} \quad (3.2.3)$$

Scrivendo anche questa in forma matriciale ed esprimendo tutti i vettori in coordinate omogenee si ha :

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} -\frac{f}{Z_C} & 0 & 0 & 0 \\ 0 & -\frac{f}{Z_C} & 0 & 0 \\ 0 & 0 & \frac{1}{Z_C} & 0 \end{bmatrix} \begin{bmatrix} X_C \\ Y_C \\ Z_C \\ 1 \end{bmatrix} \quad (3.2.4)$$

Moltiplicando a destra e a sinistra per un fattore arbitrario di scala s , si scelga ad esempio uguale a Z_C , si ottiene:

$$\begin{bmatrix} sx \\ sy \\ s \end{bmatrix} = \begin{bmatrix} -f & 0 & 0 & 0 \\ 0 & -f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X_C \\ Y_C \\ Z_C \\ 1 \end{bmatrix} \quad (3.2.5)$$

$$\hat{x} = P_P \hat{X}_C \quad (3.2.6)$$

Infine la relazione che esiste tra le coordinate dell'immagine (x, y) e le coordinate in pixel (u, v) è:

$$u = u_0 + k_u x \quad (3.2.6)$$

$$v = v_0 + k_v y \quad (3.2.7)$$

Dove (u_0, v_0) sono le coordinate del punto principale e k_u e k_v sono l'inverso della dimensione efficace del pixel, rispettivamente lungo la direzione u e v . In forma matriciale si ha:

$$\begin{bmatrix} tu \\ tv \\ t \end{bmatrix} = \begin{bmatrix} k_u & 0 & u_0 \\ 0 & k_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} sx \\ sy \\ s \end{bmatrix} \quad (3.2.8)$$

$$\hat{w} = P_C \hat{x} \quad (3.2.9)$$

A questo punto basta sostituire nell'equazione 3.2.9 la \hat{x} ricavata nella 3.2.6 per passare dalle

coordinate della scena nel sistema standard (camera) alle coordinate in pixel dell'immagine:

$$\hat{w} = P_C P_P \hat{X}_C \quad (3.2.10)$$

Sostituendo nella 3.2.10 la \hat{X}_C calcolata nella 3.2.2 si ha:

$$\hat{w} = P_C P_P P_R \hat{X} \quad (3.2.11)$$

Fattorizzando la $P_C P_P$ in un'unica matrice P_{CP} :

$$\hat{w} = P_{CP} P_R \hat{X} = \begin{bmatrix} -fk_u & 0 & u_0 & 0 \\ 0 & -fk_v & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \hat{X} \quad (3.2.12)$$

In seguito sono riportate alcune fattorizzazioni semplificative; si ponga:

$$P_{CP} = K[I \mid 0] \quad (3.2.13)$$

con

$$K = \begin{bmatrix} -fk_u & 0 & u_0 \\ 0 & -fk_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.2.14)$$

$$P_R = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} \quad (3.2.15)$$

con

$$R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \quad (3.2.16)$$

$$t = \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} \quad (3.2.17)$$

In base alle fattorizzazioni fatte si può scrivere:

$$\hat{w} = K[I | 0] \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} \hat{X} = K[R | t] \hat{X} = P \hat{X} \quad (3.2.18)$$

con P :

$$P = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{bmatrix} \quad (3.2.19)$$

Dove P è la *matrice di proiezione prospettica (MPP)* e rappresenta il modello geometrico della telecamera. Essa è una matrice 3×4 e dipende da 11 parametri indipendenti a meno di un fattore di proporzionalità, dato che si utilizzano vettori espressi in coordinate omogenee. Per poter calibrare la telecamera sono necessari solo 6 punti dato che ogni punto è identificato da due equazioni, quindi 6 punti sono identificati da 12 equazioni con 11 incognite. In ogni caso, per minimizzare il problema dell'incertezza di misura delle coordinate del punto conviene utilizzarne un maggior numero possibile.

La matrice $[R|t]$ contiene sia la rotazione che la traslazione del sistema mondo rispetto a quello della telecamera; si dice che tale matrice contiene i *parametri estrinseci* della telecamera che descrivono la posizione e l'orientamento della telecamera rispetto ad un sistema di riferimento prestabilito

(sistema mondo). Infine, nella matrice K si può porre $a_u = -fk_u$ $a_v = -fk_v$ che

rappresentano rispettivamente la lunghezza focale espressa in pixel orizzontali e verticali. Queste ultime insieme a u_0 e v_0 rappresentano i *parametri intrinseci* della telecamera che descrivono la geometria interna e le caratteristiche ottiche della telecamera.

I parametri estrinseci ed intrinseci caratterizzano completamente la telecamera; infatti, la calibrazione si può definire come quel processo che permette di ricavare le matrici di tali parametri [13].

3.3 Intrinsic Calibration

La calibrazione si basa sulla misura in modo accurato dei parametri intrinseci ed estrinseci del modello della telecamera. Per primo si calcolerà la matrice P (matrice di proiezione prospettica) e successivamente da questa si potranno estrarre i parametri della telecamera tramite decomposizione. E' necessario munirsi di un oggetto di calibrazione, che nel nostro caso è rappresentato da una scacchiera di dimensione 960mm x 56mm con quadretti di colore alterno e di lato 80mm.



Figura 3.3.1: Oggetto di calibrazione

La calibrazione richiede precisione nelle misure iniziali e una fase di ottimizzazione dei risultati. Tale lavoro risulta essere fortemente condizionato dalla qualità dell'ottica del dispositivo di acquisizione.

I seguenti punti sono i passi fondamentali per effettuare la calibrazione della telecamera :

- stampare e fissare un modello su di una superficie planare;
- acquisire da varie angolazioni una serie di immagini del calibro;
- rilevare i punti che caratterizzano il calibro nell'immagine;
- valutare i parametri intrinseci ed estrinseci;
- valutare i coefficienti della distorsione radiale;
- effettuare la minimizzazione dei parametri.

Nell'applicazione pratica per effettuare la calibrazione è stato utilizzato il Camera Calibration ToolBox, una scacchiera e le telecamere del 3MORDUC considerate singolarmente. Le sequenze di immagini utilizzate per una delle due calibrazioni sono illustrate in Figura 3.3.2.

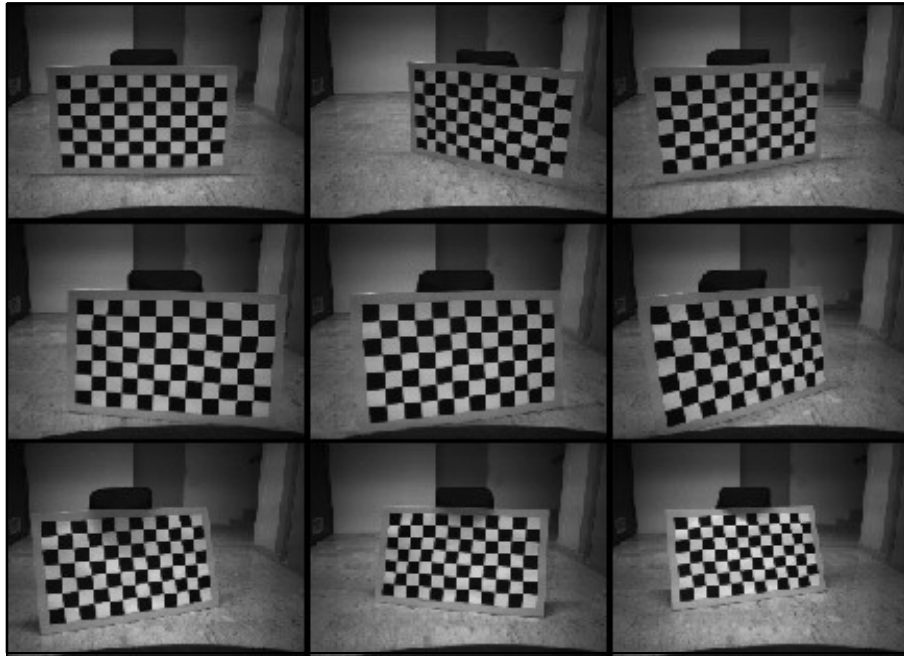


Figura 3.3.2: Immagini di calibrazione telecamera sinistra

Per descrivere il procedimento seguito prendiamo in considerazione la sequenza di immagini della camera sinistra, poichè per entrambe le sequenze il procedimento è invariato, e descriviamo singolarmente le varie fasi.

La prima operazione da compiere è il caricamento delle immagini nel software seguita dall'estrazione dei corner tramite un procedimento basato su :

- digitazione del valore di *wintx*, *winty* che rappresentano le dimensioni della finestra nella direzione x ed y entro la quale l'algoritmo andrà a cercare gli angoli a partire dal punto cliccato;
- selezione automatica del conteggio dei quadretti;
- selezione dei corner estremi del pattern di calibrazione per ogni immagine caricata;
- digitazione della lunghezza in millimetri dei quadretti lungo l'asse x e lungo l'asse y.

Dopo aver eseguito le operazione sopra esposte, vengono calcolati dal sistema i corner interni all'area selezionata dall'utente e visualizzati sull'immagine di partenza come delle croci rosse, così come viene mostrato in Figura 3.3.3.

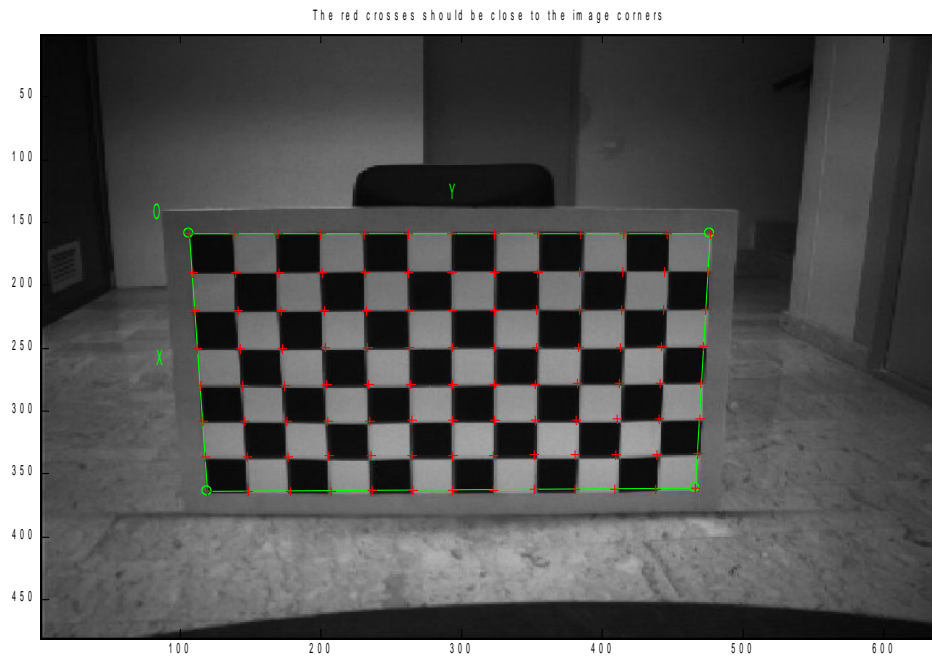


Figura 3.3.3: visualizzazione corner

Infine viene mostrata una nuova immagine dove vengono evidenziati i corner che sono stati estratti dall'algoritmo (Figura 3.3.4)

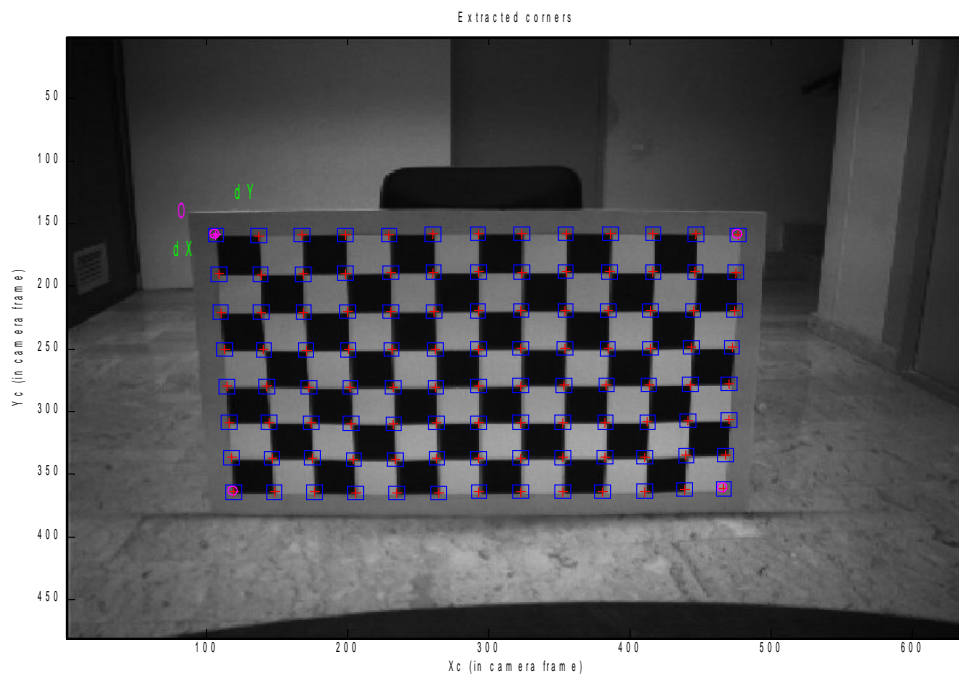


Figura 3.3.4: visualizzazione corner estratti dall'algoritmo

Questo procedimento viene ripetuto per tutte le immagini scelte per compiere la calibrazione.

A questo punto si procede ad una prima stima dei parametri della telecamera.

Questo algoritmo si divide in due passi:

- **inizializzazione:** viene calcolata una prima soluzione in maniera veloce ma poco accurata, senza tenere conto dei coefficienti di distorsione delle lenti;
- **ottimizzazione non lineare:** a partire dalla soluzione del primo passo si applicano una serie di iterazioni al fine di minimizzare l'errore tra i corner selezionati e quelli calcolati in base alla soluzione corrente.

Il risultato della fase di inizializzazione fornisce una prima stima dei parametri della camera:

- **Focal length(fc):** vettore 2x1 che esprime la lunghezza focale in pixel;
- **Principal point(cc):** vettore 2x1 che rappresenta il punto principale. In questa prima fase viene scelto in modo arbitrario al centro dell'immagine;
- **Skew (alpha_c):** coefficiente che definisce l'angolo tra gli assi x e y; viene scelto pari a 90 gradi;
- **Distortion(Kc):** vettore 5x1 che contiene i coefficienti della distorsione radiale e tangenziale.

Come previsto inizialmente vengono posti tutti uguali a zero. Successivamente si attiva la procedura di ottimizzazione non lineare che consiste nell'eseguire una serie di iterazioni al fine di minimizzare l'errore. L'analisi e la correzione degli errori vengono effettuate attraverso due funzioni del toolBox: *"Recomp.corners"* e *"Analyse error"*. Attraverso quest'ultima funzione è possibile visualizzare un grafico che illustra gli errori di proiezione dei punti rappresentati attraverso delle croci di differente colore, ogni colore si riferisce ad una differente immagine usata per la calibrazione. L'obiettivo è quello di avere la maggior parte delle croci nel centro del grafico che corrisponde ad avere un errore prossimo allo zero. Per cercare di raggiungere tale obiettivo può essere utilizzata la funzione *"Recomp.corners"* che permette di estrarre nuovamente i corner dalle immagini in modo automatico. Dopo aver eseguito tale funzione viene nuovamente eseguita la calibrazione. Confrontando i parametri ottenuti (Figura 3.3.5) con quelli precedenti all'estrazione automatica dei corner (Figura 3.3.6) si può osservare una lieve diminuzione del vettore *Pixel error*. I due grafici nei quali è rappresentato l'errore di proiezione dei punti sono visualizzati in Figura 3.3.7 e Figura 3.3.8; come si può notare in quest'ultimo grafico la maggior parte delle croci sono concentrate nell'intorno dell'origine e ciò è quello che volevamo ottenere.

Calibration results after optimization (with uncertainties):

```
Focal Length:      fc = [ 577.77594   576.43239 ] ± [ 4.93671   4.71649 ]
Principal point:    cc = [ 301.61238   235.87477 ] ± [ 4.48697   3.04017 ]
Skew:              alpha_c = [ 0.00000 ] ± [ 0.00000 ] => angle of pixel axes = 90.00000 ± 0.00000 degrees
Distortion:         kc = [ -0.21241   0.17405   0.00169   0.00017   0.00000 ] ± [ 0.01406   0.04745   0.00111
                        0.00122   0.00000 ]
Pixel error:        err = [ 0.41537   0.32218 ]
```

Figura 3.3.5: Risultato della calibrazione prima di aver eseguito la funzione *"Recomp.corner"*.

calibration results (with uncertainties):

```
Focal Length:      fc = [ 577.77594  576.43239 ] ± [ 4.93671  4.71649 ]
Principal point:    cc = [ 301.61238  235.87477 ] ± [ 4.48697  3.04017 ]
Skew:              alpha_c = [ 0.00000 ] ± [ 0.00000 ] => angle of pixel axes = 90.00000 ± 0.00000 degrees
Distortion:         kc = [ -0.21241  0.17405  0.00169  0.00017  0.00000 ] ± [ 0.01406  0.04745
                        0.00111  0.00122  0.00000 ]
Pixel error:        err = [ 0.41530  0.32153 ]
```

Figura 3.3.6: Risultato della calibrazione dopo aver eseguito la funzione “Recomp.corner”.

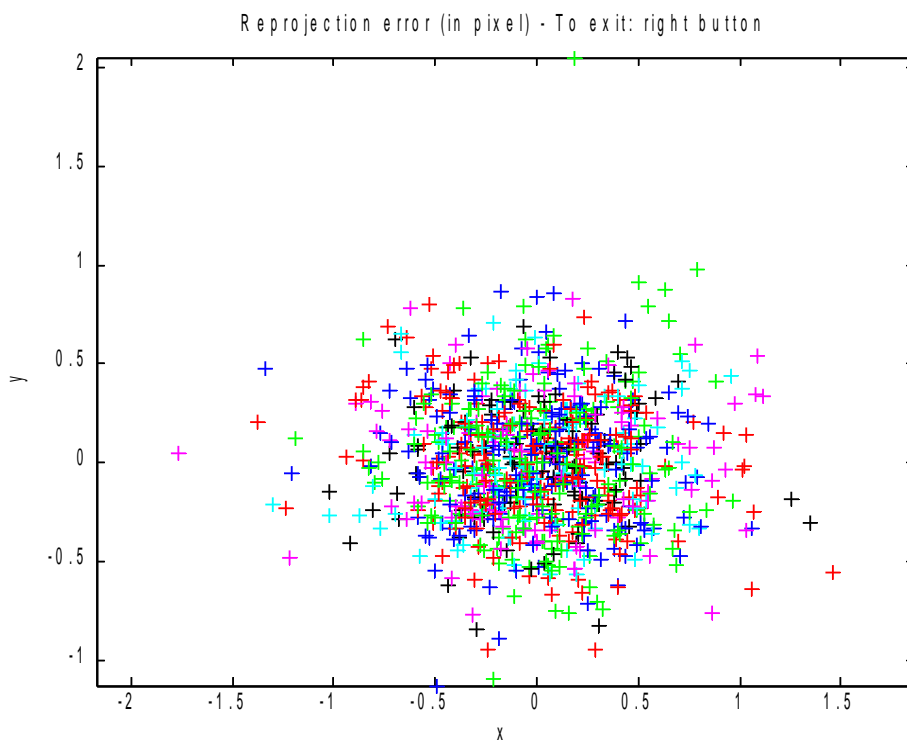


Figura 3.3.7: Errore di proiezione dei punti prima di aver eseguito la funzione “Recomp.corner”.

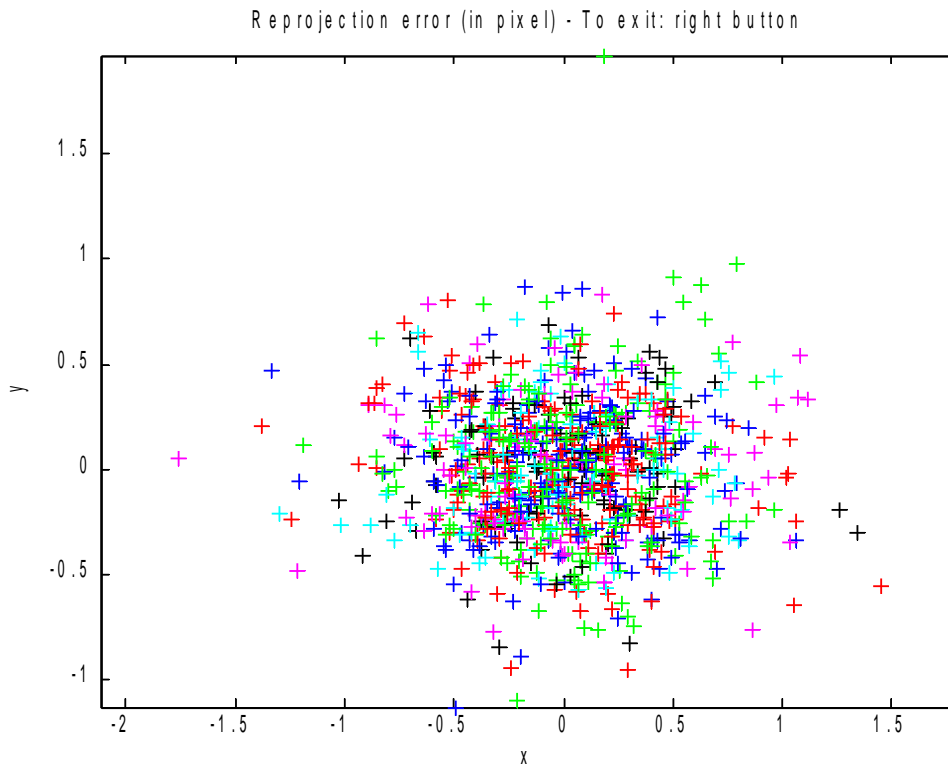


Figura 3.3.8: Errore di proiezione dei punti dopo aver eseguito la funzione "Recomp.corner".

Le telecamere inoltre sono affette da alcune distorsioni geometriche. Nei sistemi ottici composti da lenti sferiche, con centro lungo l'asse ottico, tali distorsioni si avvertono lungo la direzione radiale. In pratica un punto viene proiettato ad una distanza dal punto principale che è maggiore o minore di quella predetta dalle equazioni della proiezione prospettica. Tale distanza aumenta man mano che ci si allontana dal centro. E' possibile tenere conto della distorsione radiale attraverso le seguenti due equazioni:

$$\delta_x = x(k_1 r^2 + k_2 r^4 + ..) \quad (3.3.1)$$

$$\delta_y = y(k_1 r^2 + k_2 r^4 + ..) \quad (3.3.2)$$

dove x e y sono misurate dal centro della distorsione, che tipicamente è il punto principale, e la grandezza r è la distanza dal punto principale.

Un altro tipo di distorsione di cui si può tenere conto è la distorsione tangenziale. Anche questa cresce all'aumentare della distanza dal centro e si può modellare come:

$$\delta_x = -y(\varepsilon_1 r^2 + \varepsilon_2 r^4 + ..) \quad (3.3.3)$$

$$\delta_y = x(\varepsilon_1 r^2 + \varepsilon_2 r^4 + ..) \quad (3.3.4)$$

Utilizzando il toolbox è stato possibile visualizzare tali distorsioni mandando in esecuzione lo script *visualize_distortions*. I risultati ottenuti sono illustrati in Figura 3.3.9, Figura 3.3.10 e Figura 3.3.11. In particolare nella prima figura viene mostrato il modello della distorsione completo, ovvero quello ottenuto combinando l'effetto della distorsione radiale e tangenziale. In essa la croce rappresenta il centro effettivo dell'immagine, il cerchio il punto principale ed ogni freccia indica l'effettivo spostamento di un pixel indotto dalla distorsione delle lenti. Nella seconda figura, invece, viene rappresentata la componente tangenziale della distorsione. Nella terza figura, infine, viene mostrata la componente radiale della distorsione abbastanza simile al grafico del modello completo.

Alessandro Mazzurco

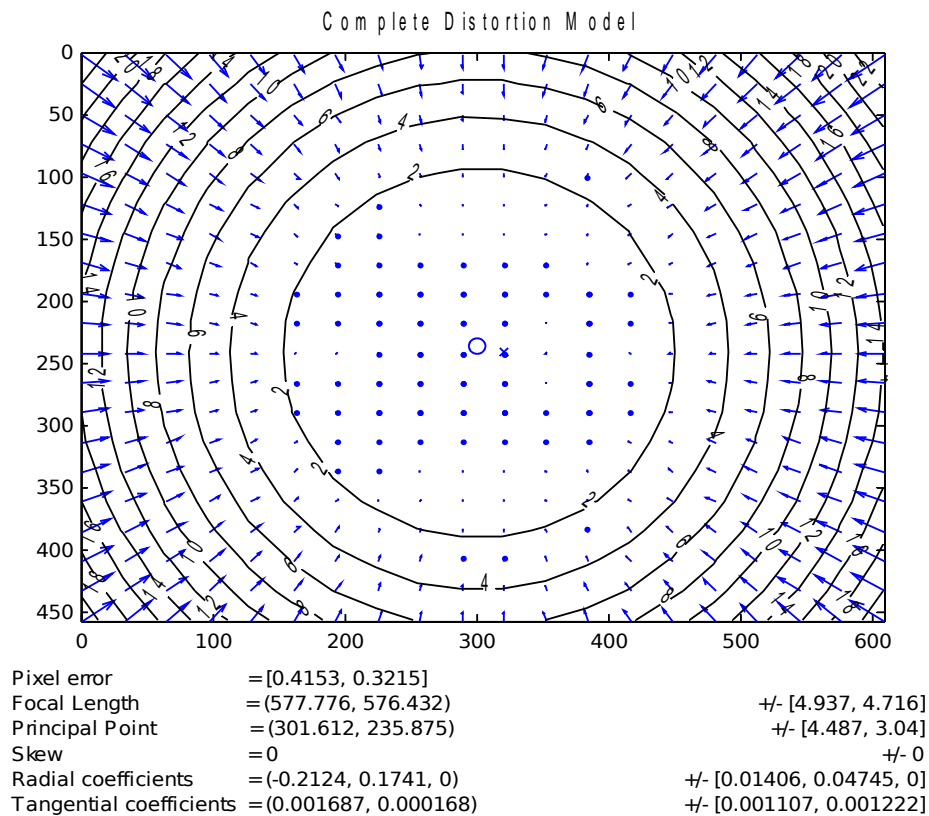


Figure 3.3.9: Complete Distortion Model

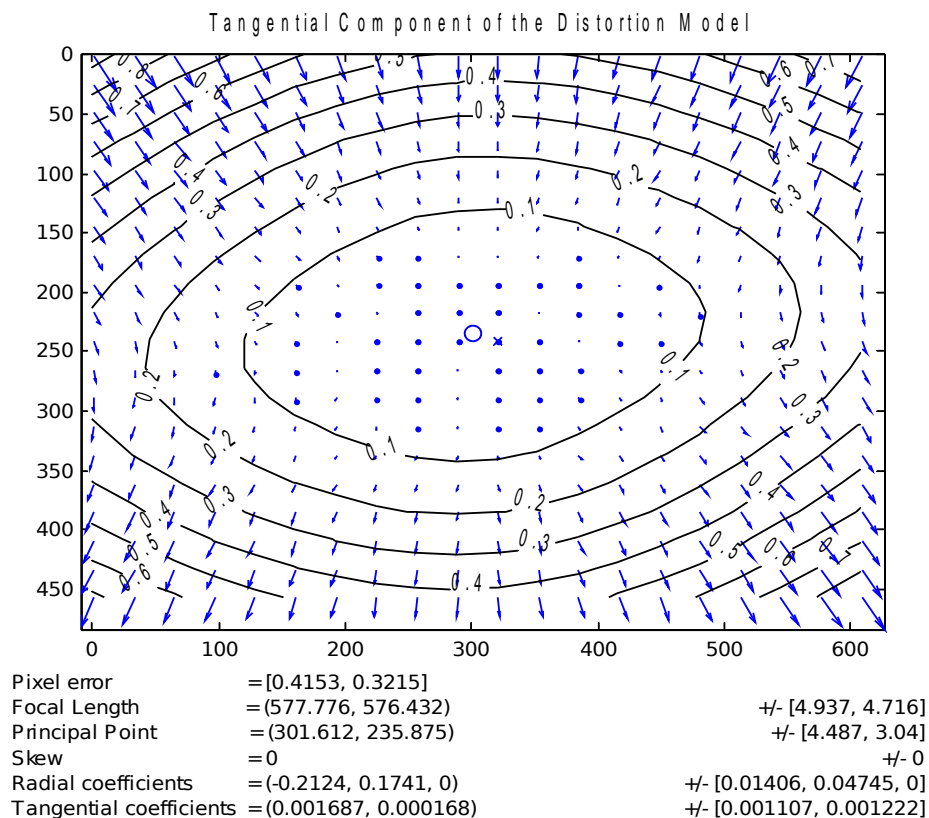


Figure 3.3.10: Tangential Component of the Distortion Model

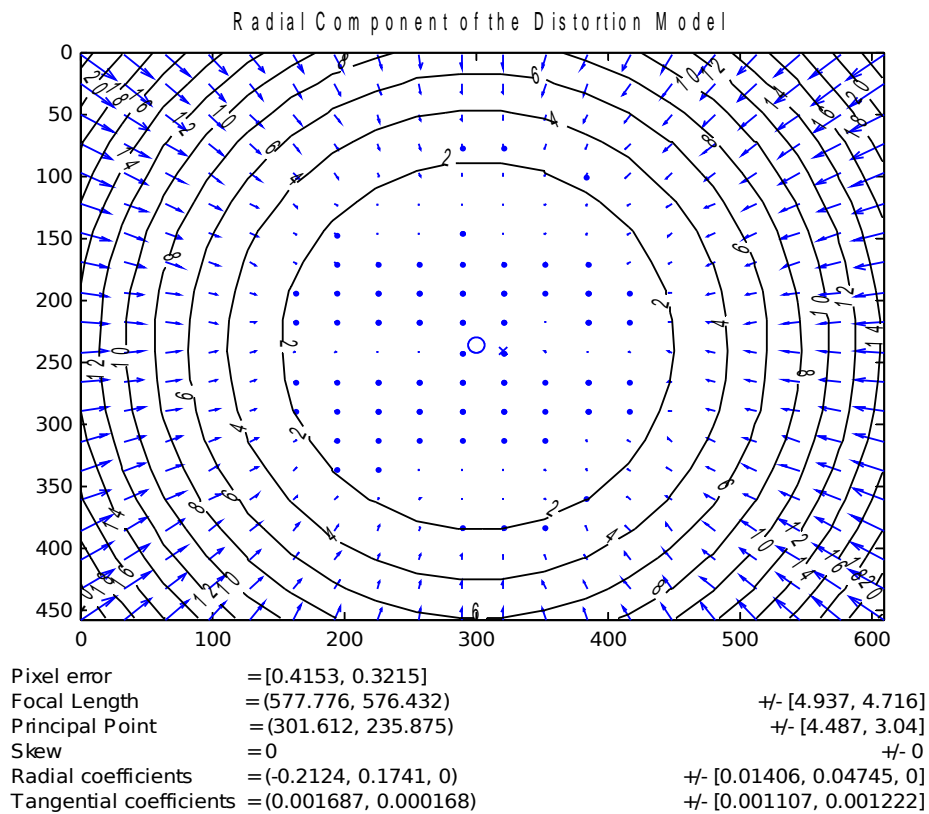


Figura 3.3.11: Radial Component of the Distortion Model

Per eliminare le aberrazioni indotte dalla distorsione si è utilizzata la funzione “*Undistort image*” applicata su tutte le immagini che ne ha permesso la loro rettificazione. Il risultato di tale funzione è mostrato in Figura 3.3.12.

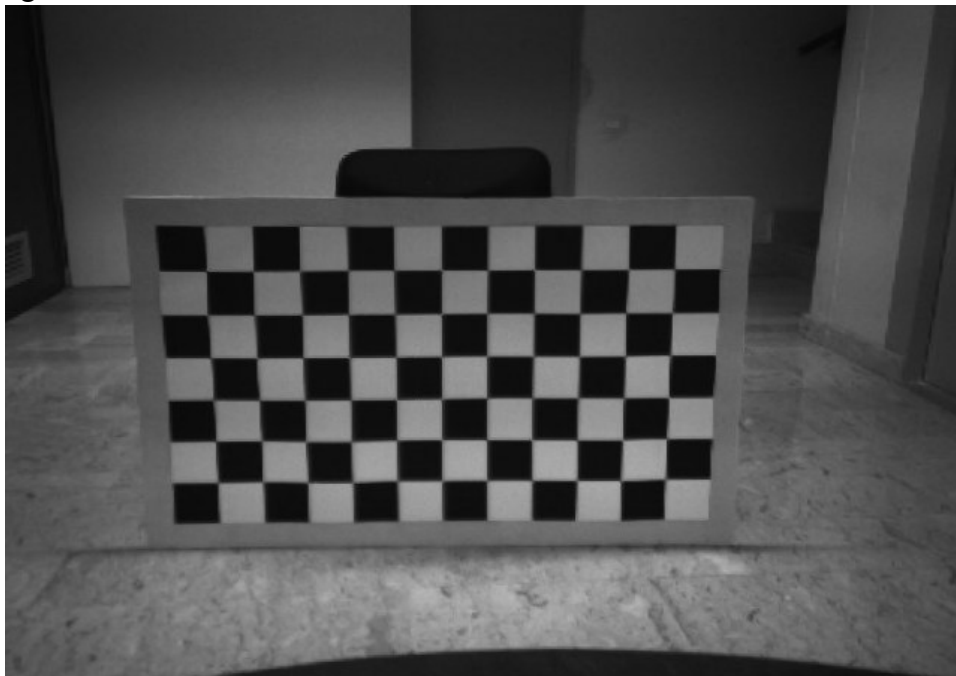


Figura 3.3.12: immagine rettificata

Attraverso questo toolBox è anche possibile calcolare i parametri estrinseci di una qualsiasi immagine dati i parametri intrinseci ricavati attraverso la precedente calibrazione. In particolare, i

parametri estrinseci sono codificati in una matrice di rotazione (Rc_ext), in un vettore di traslazione (Tc_ext) ed in un vettore di rotazione omc_ext relativo alla matrice di rotazione. Utilizzando i parametri estrinseci possiamo ricavare la corrispondenza tra un punto dello spazio ed un pixel dell'immagine e viceversa. Quindi, dato un punto P dello spazio di coordinate $XX=[X;Y;Z]$ si può ricavare il corrispondente pixel nella telecamera attraverso l'equazione:

$$XX_C = RC_ext * XX + Tc_ext \quad (3.3.5)$$

dove $XX_C=[X_C;Y_C;Z_C]$ è rappresentato dalle coordinate del corrispondente punto nel sistema della telecamera.

Per il nostro obiettivo non è stato necessario ottenere mediante tale tool i parametri estrinseci delle singole telecamere che sono stati valutati successivamente nel secondo step, che prevede la calibrazione estrinseca tra il laser e camera. Tuttavia di seguito sono riportate due figure in cui vengono visualizzate rispettivamente le diverse posizioni della camera assumendo che il calibro (quindi il sistema mondo) sia fisso, e viceversa le diverse angolazioni assunte dal calibro, ipotizzando che il sistema della camera sia fisso (come avviene realmente).

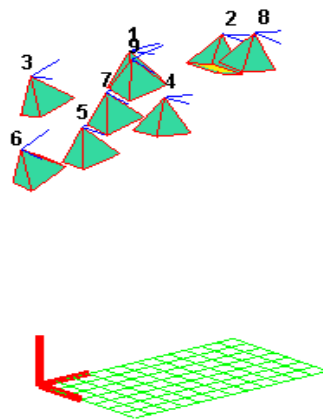


Figura 3.3.13: Sistema mondo.

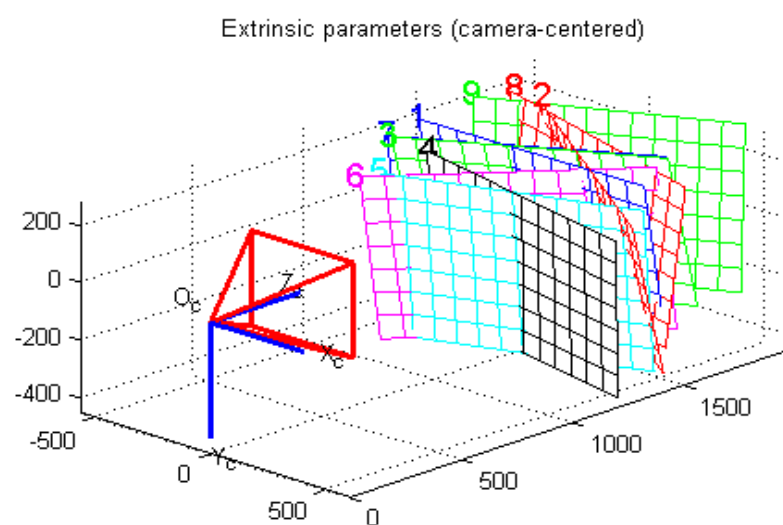


Figura 3.3.14: Sistema camera.

3.4 Extrinsic Calibration

Il procedimento riportato in questo paragrafo e le relative formule si basano sull'articolo menzionato nel paragrafo 2.2 di questo report.

La calibrazione estrinseca laser-camera consiste nella valutazione della matrice di rotazione Φ e del vettore traslazione Δ che permettono la trasformazione rigida di un qualsiasi punto dello spazio espresso nel sistema di coordinate laser al sistema di coordinate camera (vedi formula 2.2.2). A tale fine viene impiegato un pattern piano di calibrazione visualizzato simultaneamente dal laser e dalla camera. Per ogni differente posizione del pattern, il metodo vincola i termini della matrice di rototraslazione (ovvero della matrice Φ e del vettore Δ) attraverso i dati laser acquisiti sul pattern e l'immagine proveniente dalla camera.

Come abbiamo visto precedentemente una proiezione dalle coordinate reali $P=[X, Y, Z]^T$ alle coordinate immagini $p=[u, v]^T$ può essere rappresentata come segue:

$$p \simeq K(RP + T) \quad (3.4.1)$$

dove K è la matrice dei parametri intrinseci, R è una matrice 3×3 ortonormale che rappresenta l'orientamento della camera e T è un vettore 3×1 che rappresenta la posizione della camera. Nel caso reale bisogna anche considerare un altro vettore 5×1 che chiamiamo Kc e modella la distorsione delle lenti. In questa fase di calibrazione laser-camera i parametri intrinseci K e il vettore distorsione Kc vengono considerati noti e proprio per questo la calibrazione laser-camera è successiva alla calibrazione della camera.

Noti i parametri intrinseci ed il vettore distorsione entrambi costituiscono l'input di questo step assieme al report laser, contenente le 181 distanze misurate nell'arco di 180° in un piano parallelo al terreno $Y=0$, ed alle immagini provenienti da camera.

Allo scopo quindi di determinare la trasformazione 2.2.2 si assume senza, perdere di alcuna generalità, che il pattern (o equivalentemente calibro, checkboard) stia sul piano $Z=0$ nel sistema di coordinate mondo. Nel sistema di coordinate camera allora il pattern può essere parametrizzato da un vettore 3×1 parallelo alla normale al calibro e che chiameremo N . Il modulo di N è uguale alla distanza dalla camera del calibro e quindi è possibile esprimere mediante la seguente relazione:

$$N = -R_3(R_3^T T) \quad (3.4.2)$$

dove R_3 è la terza colonna della matrice R di rotazione e T è il centro della telecamera nel sistema mondo. A partire dai dati laser giacenti sul calibro che vengono dunque rilevati anche dalla camera è possibile imporre i vincoli geometrici. Considerato un punto laser P_L è possibile determinare il corrispettivo punto sul frame attraverso la seguente relazione:

$$P_c = \Phi^{-1}(P_L - \Delta) \quad (3.4.3)$$

e poiché il punto P_c sul calibro è definito da N allora esso soddisfa che:

$$N \cdot P_c = \|N\|^2 \quad (3.4.4)$$

si ottiene:

$$N \cdot \Phi^{-1}(P_L - \Delta) = \|N\|^2 \quad (3.4.5).$$

Noto P_L dalla lettura laser, valutato N come soprascritto è possibile dunque imporre il vincolo 3.4.5. E' possibile riscrivere dunque la 3.4.5 come segue:

$$N H P_L = \|N\|^2 \quad (3.4.6)$$

dove H è:

$$H = \Phi^{-1} \begin{pmatrix} 1 & 0 \\ 0 & 0 - \Delta \\ 0 & 1 \end{pmatrix} \quad (3.4.7)$$

matrice di trasformazione dalle coordinate laser a quelle camera. Per ogni posizione del calibro si ha un sistema di equazioni lineari in cui le incognite sono i termini di H , tale sistema può essere risolto attraverso lo standard metodo dei minimi quadrati. Determinata H è possibile valutare una prima soluzione lineare della posizione e dell'orientamento attraverso :

$$\Phi = [H_1 \quad -H_1 x H_2 \quad H_2]^T \quad (3.4.8)$$

$$\Delta = -[H_1 \quad -H_1 x H_2 \quad H_2]^T H_3 \quad (3.4.9)$$

dove H_i è la colonna i -esima della matrice H .

La matrice Φ ottenuta potrebbe non avere le regolari proprietà di una matrice di rotazione. In tal caso una nuova matrice $\overset{x}{\Phi}$ viene calcolata a partire da quella di partenza mediante la minimizzazione della norma di Frobenius della differenza:

$$\overset{x}{\Phi} - \Phi \text{ soggetta a } \overset{x}{\Phi} \overset{x}{\Phi}^T = I.$$

La soluzione lineare così calcolata viene utilizzata come soluzione iniziale per la successiva fase. Tale fase consta in una ottimizzazione non lineare, fisicamente più significativa, che migliora la soluzione iniziale basandosi sulla distanza euclidea dai punti laser al piano $Z=0$ del calibro.

Date le differenti posizioni del calibro la soluzione si può migliorare attraverso il funzionale sotto riportato :

$$\sum_i \sum_j \left(\frac{N_i}{N_j} \Phi^{-1} ((P_{ij} - \Delta) - \|N\|^2) \right) \quad (3.4.10)$$

Dove N_i definisce il calibro nella i -esima posizione e Φ è il vettore 3×1 che parametrizza la matrice di rotazione mediante la nota formula di Rodrigues. Tale funzionale è minimizzata attraverso Lavenberg-Marquardt Algorithm in Matlab.

Poiché sia laser che camera hanno entrambi rumori in uscita, tale procedimento prevede un metodo robusto iterativo con lo scopo di ottenere una soluzione convergente con il più basso residuo possibile.

Tale metodo si basa nei seguenti passi:

1. per ogni posizione del calibro viene valutato il vettore N ed estratti i punti laser giacenti sul calibro;
2. basandosi sulla corrente posizione del calibro si determinano Φ e Δ attraverso la

Alessandro Mazzurco

minimizzazione di una versione pesata della 3.4.10, dove il peso viene determinato dalla stima del rumore dei punti laser;

3. per ogni posizione si valuta la media dell'errore di riproiezione dei punti laser sul calibro della corrente immagine. Se tale errore è minore di un δ threshold, errore medio massimo, allora tale posizione è considerata valida;

4. si ripetono i punti 2 e 3 fino alla convergenza.

L'intero procedimento sopra descritto è stato elaborato ed implementato in Matlab da Q. Zhang e R. Pless. Gentilmente gli autori forniscono una versione di tale codice Matlab che è stata da me utilizzata con lo scopo di ottenere una calibrazione laser camera e quindi il mapping dei punti laser sull'immagine proveniente da ogni singola camera della stereocamera del 3Morduc.

I compiti da me eseguiti per potere effettuare tale calibrazione sono riassunti nei seguenti punti:

- costruzione di un calibro 96x56 cm con quadrati di 8 cm (12x8 quadrati);
- acquisizione delle immagini e dei dati laser da 3Morduc mediante collegamenti con Catania;
- split delle immagini provenienti da stereocam, in due singole immagini e scelta delle immagini da utilizzare;
- valutazione dei parametri intrinseci della camera mediante il ToolBox Calib;
- valutazione della matrice Φ e del vettore Δ mediante il Tool di funzioni Zhang & Pless.

Una volta acquisite le immagini e valutati i parametri intrinseci di una delle due camere, si procede con il *Tool Zhang & Pless*. A tale Tool vengono forniti, come già scritto sopra, la matrice dei parametri intrinseci ed il vettore distorsione direttamente sul workspace di Matlab. Successivamente si utilizza la funzione *extCalib* dal command window che prende in ingresso proprio i parametri intrinseci e di distorsione e fornisce in uscita la matrice Φ ed il vettore Δ desiderati.

Nel nostro caso è stato rilevato un errore Matlab per cui è risultato impossibile utilizzare tale funzione. Tale errore potrebbe essere causato dall'aver utilizzato un calibro leggermente diverso rispetto a quello per cui il codice è stato elaborato. Nel calibro da me realizzato non sono presenti dei cerchi concentrici lungo i 4 angoli, utili al fine di rilevare il pattern nell'immagine, estrarre i corners e valutare i parametri estrinseci della camera per ogni ciascuna immagine. Una verifica della suddetta considerazione è stata fatta attraverso una immagine modificata con photoshop in cui sono stati ritoccati gli angoli del calibro rendendolo molto simile a quello usato nel sample.



Figura 3.4.1: Calibro ritoccato con photoshop.

Purtroppo questa modifica non è stata di alcun ausilio.

Alessandro Mazzurco

Per tale motivo è stata utilizzata una funzione equivalente, *extCalibManual*, in cui è necessario, per ogni immagine, cliccare sui quattro corners del calibro i quali vengono così riconosciuti come estremità di una regione all'interno del quale individuare i rimanenti corners ed effettuare la calibrazione estrinseca.(Figura 3.4.1).

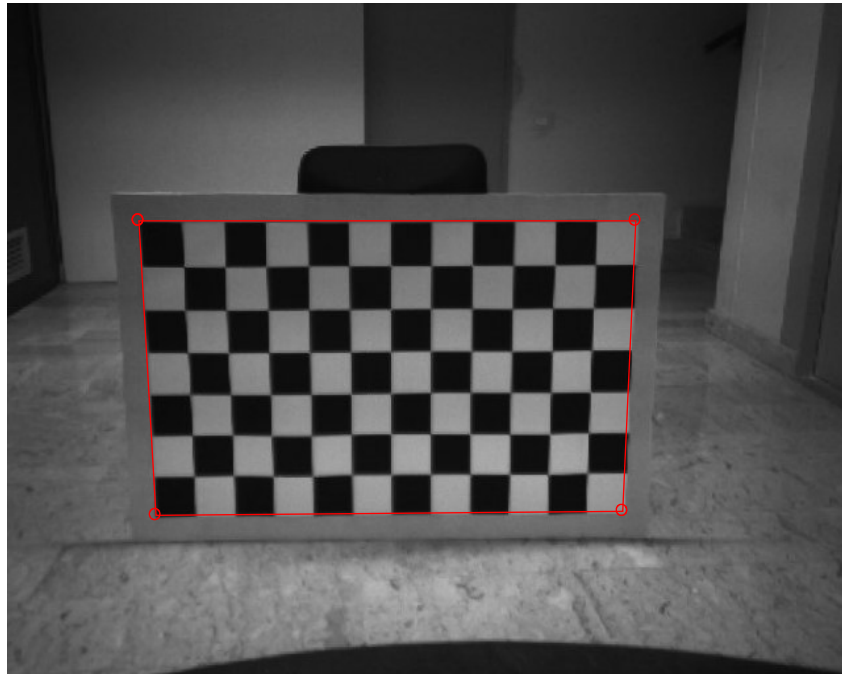


Figura 3.4.2: In rosso la regione del calibro in cui saranno estratti i corners

Nonostante l'utilizzo di questa versione manuale i risultati ottenuti sono stati inaccettabili in quanto tutti i punti laser venivano mappati in un range molto ristretto di pixel (2-3 pixel contengono tutti i punti laser).

Questo mi ha condotto ad una fase di analisi dettagliata del codice che mi ha permesso di individuare una funzione interna *getLinePts*. Tale funzione ha come scopo quello di determinare i punti laser presenti sul calibro. Questa rilevazione viene compiuta automaticamente oppure tramite una modalità interattiva che permette di operare sui dati laser. Attraverso quest'ultima modalità, è possibile selezionare tramite un click i due estremi del calibro sulla mappa laser, espressa in un sistema cartesiano. Il risultato di questa funzione viene stampato a video anche attraverso un plot che è riportato in Figura 3.4.3, dove in rosso vengono indicati i punti laser presenti sulla scacchiera.

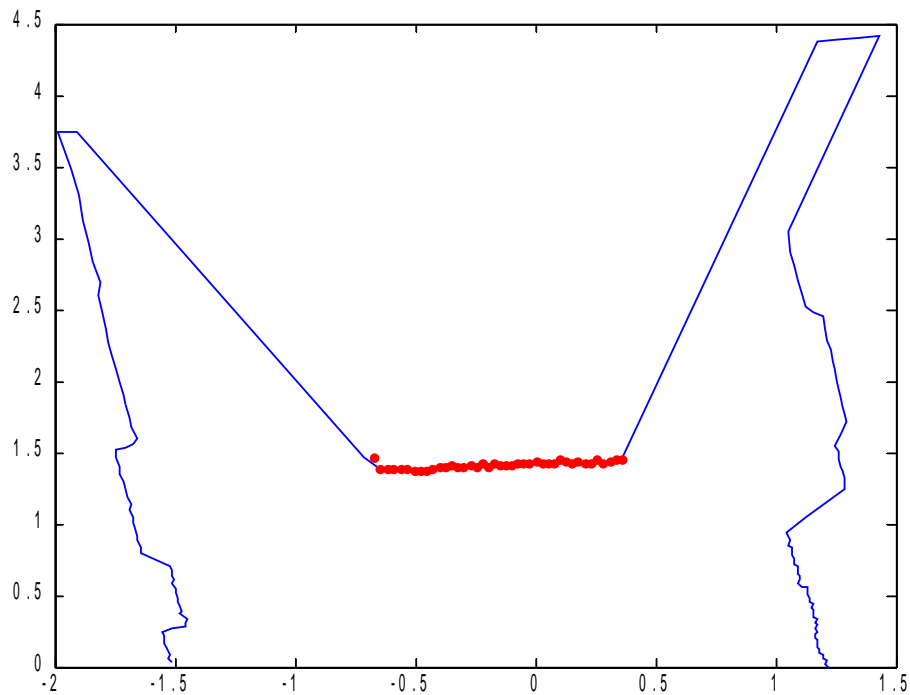


Figura 3.4.3: Mappa laser in un sistema cartesiano con origine nel centro del laser; in rosso sono riportati i punti sul calibro.

Sono state effettuate delle prove anche con la modalità interattiva senza però nessun esito positivo. Altre prove sono state effettuate mediante una funzione da me realizzata *punticamera* con lo scopo di mappare i punti laser correttamente sull'immagine. Questa funzione effettua una trasformazione dei dati laser acquisiti in coordinate polari in un sistema cartesiano e successivamente opera la trasformazione 3.4.3. Per finire colora di rosso i pixel presenti nell'immagine. Questa prova non ha risolto il problema riscontrato anche utilizzando la funzione *mapping* presente nel Tool Zhang & Pless. Questo è indice che il problema nasce durante la fase di processamento della funzione *extCalibManual* e non durante il mapping.

All'interno della funzione *extCalibManual* vengono calcolati per ogni singola immagine i parametri estrinseci della camera al fine di valutare il vettore N che parametrizza il calibro. Questa fase può essere effettuata anche con il *Toolbox Calib* precedentemente usato per valutare i parametri intrinseci. Proprio per questo ho pensato come possibile escamotage quello di computare precedentemente i parametri estrinseci della camera per ogni immagine e darli in input alla funzione *getCamExParams* interna a *extCalibManual* evitando così la procedura manuale di rilevamento degli angoli. Tutto questo è stato fatto attraverso una funzione leggermente modificata chiamata *myextCalibManual* che non ha alcun parametro in ingresso, in quanto i parametri intrinseci e la distorsione erano utili solo allo scopo della computazione dei parametri estrinseci. Nonostante ciò i risultati ottenuti sono rimasti invariati. Un'ulteriore prova senza esito positivo è stata effettuata anche attraverso le immagini corrette da distorsione mediante il *Toolbox Calib* descritto nel precedente paragrafo.

In virtù di tutte le prove eseguite presumo che il problema di tale risultato risieda nella funzione *getTransformMatrix* dove viene minimizzato il funzionale pesato presente nella 3.4.10.

Probabilmente le posizioni assunte dal calibro durante l'acquisizione delle immagini dal 3Morduc potrebbero essere un'ulteriore causa del problema riscontrato. Infatti il calibro dovrebbe essere posto sempre alla medesima distanza dalla camera, variando esclusivamente le inclinazioni col terreno e l'angolo di rotazione attorno all'asse perpendicolare alla terra.

Alessandro Mazzurco

Di seguito riporto due immagini del calibro su cui vengono mappati non correttamente i punti laser e uno zoom della stessa allo scopo di visualizzare i risultati ottenuti.

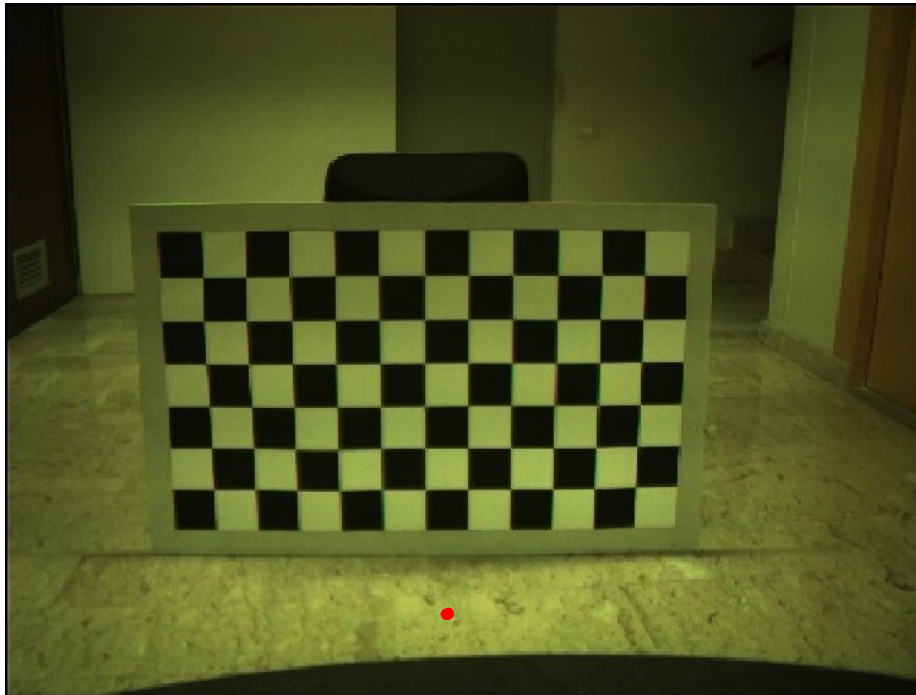


Figura 3.4.3: Immagini con punti laser mappati non correttamente

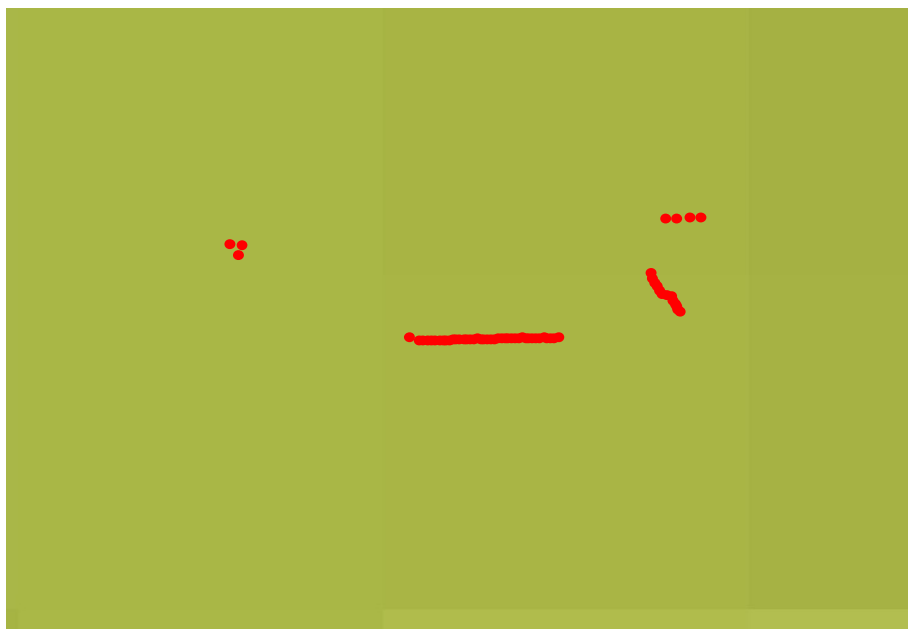


Figura 3.4.4: Zoom dell'immagine precedente.

4 CONCLUSIONI

I risultati ottenuti in questo lavoro nonostante non sia riuscito ad realizzare l'obiettivo prefissato mi hanno permesso di approfondire le tematiche e i procedimenti da attuare per calibrare un sistema sensoriale come quello presente nel 3Morduc. Tuttavia tale lavoro necessita di ulteriore tempo uomo per potere realizzare l'obiettivo in questo report descritto.

Ritengo che i problemi da me riscontrati, certamente di natura tecnica, richiedano solamente una nuova analisi. Credo che la strada da me intrapresa possa essere una possibile soluzione per la sensor fusion sul 3Morduc e quindi un modo per potere migliorare la teleguida di tale robot.

Per perseguire tale fine continuerò a sviluppare questo lavoro dopo la conclusione del tirocinio.

Dal punto di vista umano tale esperienza mi ha permesso di avvicinarmi al mondo del lavoro, di inserirmi in un contesto di teamwork in ambito internazionale e di integrarmi nella compagine europea.

Ringraziamenti

Ringrazio particolarmente il prof. Giovanni Muscato referente italiano di questo progetto, *Erasmus Placement*, e il prof. Salvatore Livatino, supervisore in UK, che mi ha sostenuto in questi mesi di lavoro con pazienza e vigore.

APPENDICE A

Agenda

Attraverso la seguente agenda si è impostato il lavoro svolto durante il tirocinio presso l'Università di Hertfordshire secondo una breve descrizione del lavoro pianificato settimanalmente. In particolare tale agenda è descritta più dettagliatamente a partire dal 3 Agosto.

Sotto la supervisione del prof. Salvatore Livatino sono stati stabiliti alcuni macrotasks di seguito riportati:

- approfondimento della letteratura riguardante sensor fusion, matching tra bitmap da camera e dati laser: pubblicazioni e tesi precedenti;
- valutazione e scelta dell'approccio più opportuno da implementare allo scopo di sovrapporre dati video e laser in un'unica modalità;
- implementazione e verifica sperimentale di eventuali successi o fallimenti e deduzioni.

Di seguito è riportata l'agenda settimanale:

11-19 Giugno

Lecture: Tesi Carmelo Arena, Alba Pennisi, Giovanni Lanteri, Report Luca&Francesco, book Learning OpenCV.

Installazione: Visual Studio 2005 C++, OpenCV, DevC++.

Prove: prova tecnica di comunicazione con 3MORDUC ed esempi di funzioni con OpenCV.

Meeting universitario: "Depth-Enhanced Bronchoscope Teleguide" di S.Livatino, A. Zocco, A. Agrimi.

22-26 Giugno

Lecture documenti: "Accurate Motion Estimation and High-Precision 3D Reconstruction by Sensor Fusion".

Lecture: Geometria Epipolare e calibrazione stereocamera.

Documentazione sul Software esistente: David 3D.

29-3 Luglio

Lecture documenti: "Extrinsic Calibration of Camera and Laser Range Finder".

Lettura funzioni in openCV per la calibrazione intrinseca ed estrinseca.

Scrittura iniziale di un tutorial con i passi essenziali per la calibrazione

6-10 Luglio

Lettura degli argomenti e funzioni in OpenCV riguardanti: Edge Detection (algoritmo di Canny), Line Detection (trasformata di Hough), Corner Detection (Harris corner detector).

Analisi delle funzioni e metodi usati nel report Lanteri per l'eliminazione delle informazioni inconsistenti.

13-17 Luglio

Scrittura prima bozza tutorial per la calibrazione estrinseca tra camera e laser.

Lettura calibrazione estrinseca della stereo-cam, ricostruzione e matching.

Prova tecnica con 3Morduc-Catania.

20-24 Luglio

Riferimenti, bibliografia, appendice, indice e introduzione.
Ricerca di toolbox esistenti per la calibrazione.
Lettura del codice Matlab per la calibrazione estrinseca.

27-31 Luglio

Costruzione pannello di prova, prove di calibrazione intrinseca ed estrinseca.
Costruzione del pannello finale.
Scrittura Report.

3-7 Agosto

Piano Settimanale proposto: calibrazione intrinseca delle due camere che compongono la stereocamera, calibrazione estrinseca del sistema sensoriale laser-camera.

Piano realizzato:

Lunedì ore 9-13, 15-19 organizzazione con Catania e analisi del codice Pless;
Martedì ore 10-16:30, 18-21 acquisizione immagini da 3Morduc di molteplici posizioni del pannello di calibrazione mediante sessione con l'Università di Catania, scelta delle immagini più opportune da utilizzare ai fini della calibrazione;
Mercoledì ore 9-13, 15-19 calibrazione intrinseca della camera sinistra e destra mediante TOOLBOX_calib e organizzazione dei risultati ottenuti;
Giovedì ore 10-14, 17-21 calibrazione estrinseca del sistema costituito da camera sinistra e laser mediante codice Matlab di Robert Pless, riscontro di problemi nella funzione *detectargets* in cui non si riescono ad individuare i corner del calibro utilizzato;
Venerdì ore 10-13, 17-21 analisi della funzione *detectargets* con lo scopo di individuare e risolvere il problema.

10-14 Agosto

Piano settimanale proposto: motivazioni sugli articoli considerati per il progetto, prove di calibrazione estrinseca con immagini ritoccate in cui vengono aggiunti i features concentrici negli angoli.

Piano realizzato:

Lunedì ore 9-13, 15-19 analisi e prove con delle immagini in cui il pannello di calibrazione ruota attorno ad un asse ortogonale al piano di terra.
Martedì ore 8:30-12, 15-18 modifica delle immagini acquisite applicando dei cerchi concentrici negli angoli del pannello di calibrazione e prove.
Mercoledì ore 10-12, 14:30-19 visione degli articoli riguardanti la calibrazione e scrittura del report in merito alle motivazioni sugli articoli scelti.
Giovedì ore 9-13, 15-18 stesura e revisione del terzo capitolo riguardante la calibrazione.
Venerdì 8-15 modifica dell'immagine del pannello di calibrazione per valutare l'eventuale risoluzione del problema con la funzione *detecttargets*. Revisione e aggiornamento del report.

17-21 Agosto

Piano settimanale proposto: risoluzione del problema di identificazione degli angoli del calibro, conclusione della calibrazione estrinseca laser-camera e verifica dei risultati ottenuti.

Piano realizzato:

Lunedì ore 11-15, 17-20 soluzione del problema di rilevazione degli angoli del calibro mediante

Alessandro Mazzurco

l'utilizzo di una funzione alternativa *extcalibmanual*.

Martedì ore 10-13, 15-18 prova dei risultati ottenuti attraverso plot in Matlab e riscontro di dati laser incongruenti con l'ambiente reale scansionato, probabilmente, a causa di fenomeni di riflessione del raggio laser sul calibro (da accertare).

Mercoledì ore 9-14, 15-17 utilizzo della funzione *getLinePts* nella sua variante interattiva che permette di selezionare i punti laser estremi su pannello.

Giovedì ore 10-13, 15-19 debug della funzione *getLinePts* che non funzionava correttamente.

Venerdì ore 10-13, 14-18 organizzazione con l'Università di Catania per una nuova sessione allo scopo di acquisire nuove immagini, ulteriori prove con le immagini possedute e plot dei dati laser (sistema polare) in un sistema cartesiano con origine nel centro del sensore laser.

24-28 Agosto

Piano settimanale proposto: stesura del terzo capitolo, acquisizione delle nuove immagini da 3MORDUC e calibrazione.

Piano realizzato:

Lunedì ore 10-14, 15-20 calibrazione estrinseca camera-laser e prove di mapping dei punti laser sull'immagine.

Martedì ore 10-14, 15-20 sessione con Catania ed acquisizione di nuove immagini dal 3Morduc, nuova calibrazione intrinseca ed estrinseca.

Mercoledì ore 9-13, 14-20 uso della funzione *mapping* presente nel “*tool Pless*” per il mapping dei punti laser e riscontro di un elevato errore nella stima della matrice di rototraslazione.

Giovedì ore 9-13, 14-19 stesura III paragrafo del terzo capitolo.

Venerdì ore 9-12, 13-19 continuazione stesura del terzo capitolo e ulteriori prove, con insuccesso, di mapping dei punti laser su immagine mediante una funzione da me realizzata (*punticamera*).

31-4 Settembre

Piano settimanale proposto: evitare l'utilizzo della funzione *argets* usando la i parametri estrinseci della calibrazione della camera, effettuare delle prove mediante le immagini corrette da distorsione, scrivere le tutte le problematiche avute nel report.

Piano realizzato:

Lunedì ore 10-13, 14-19 calibrazione intrinseca delle immagini presenti nel “*tool Pless*” allo scopo di cercare eventuali errori durante la calibrazione della camera. Da questa verifica ho ottenuto all'incirca gli stessi parametri intrinseci che riportava il sample, per cui la calibrazione della telecamera si può ritenere attendibile.

Martedì ore 10-13, 14-19 calibrazione estrinseca laser-camera dell'immagini campioni mediante la modalità manuale di riconoscimento dei 4 angoli. Da questa verifica ho riscontrato che tale funzione “*extCalibManual*” funziona correttamente.

Mercoledì ore 10-14, 15-18 valutazione dei parametri estrinseci di ogni singola immagine del 3Morduc in modo da aggirare la funzione “*getExtParameter*” in cui viene chiamata “*detectargets*” il risultato ottenuto è identico a quello che si ha senza utilizzare il seguente escamotage quindi il problema dovrebbe essere successivo alla fase di valutazione dei parametri estrinseci.

Giovedì ore 9-13, 14-18 analisi della funzione “*mapping*” e “*mapLaserPts*”, della funzione Matlab “*repmat*”, “*project_points2*” allo scopo di ottenere un mapping corretto dei punti laser su immagine.

Venerdì ore 10-17 analisi della funzione “*getTransformMatrix*” la quale restituisce la matrice di rototraslazione, utilizzo dell'*if* interattivo nella funzione “*getLinePts*” che permette di selezionare i punti laser che ricadono sul pannello, analisi della funzione Matlab “*polyfit*”.

7-11 Settembre

Piano finale proposto: scrittura del report e ulteriori prove di calibrazione.

Piano realizzato:

Lunedì ore 9-13, 15-18 documentazione e calcolo del FOV (field of view) della camera mediante i risultati ottenuti.

Martedì ore 9-12, 13-17 verifica dei passi seguiti durante la calibrazione per accertarmi di eventuali errori non riscontrati.

Mercoledì ore 10-14, 15-18 scrittura del report terzo capitolo.

Giovedì ore 10-14, 15-18 scrittura del report .

Venerdì ore 9-14, 15-18 revisione report.

14-16 Settembre

Lunedì ore 9-13, 14-17 correzione report ed integrazioni di immagini.

Martedì ore 10-13, 14-17 organizzazione della bibliografia e modifica del layout del report.

Mercoledì ore 11-13, 15-17 revisione generale del report e stampa.

BIBLIOGRAFIA

- [1] F. Catania and M. Mazzaglia. "Towards Augmented Reality Visualization in Mobile Robot Teleguide", pages 11-13, 10.
- [2] S. Livatino, G. Muscato, S. Sessa, C. Köffel, C. Arena, A. Pennisi, D. Di Mauro and A. Malkondur. "Mobile Robotic Teleguide Based on Video Images", *IEE Robotics and Automation Magazine*, 2008.
- [3] 3MO.R.D.U.C., "3rd version of the MOBILE Robot DIEES University of Catania", website: <http://www.robotic.diees.unict.it/robots/morduc/morduc.htm>.
- [4] User's Manual, "Laser scanner", website: http://www.uvs-info.com/pdf/UGV-Datasheets/Sick_Germany_2D-Laser-scanner_LMS_071116.pdf.
- [5] User's Manual, "STH-MDCS2-VAR-C", website: <http://www.videredesign.com/docs/sthmdcs2-var.pdf>.
- [6] C. Arena, Tesi di laurea, "Depth enhanced mobile robot teleguide using laser and video sensors", Università di Catania.
- [7] Yunsu Bok, Y. Hwang and In So Kweon. "Accurate Motion Estimation and High-Precision 3D Reconstruction by Sensor Fusion", *2007 IEEE International Conference on Robotics and Automation, Roma, Italy, 10-14 April 2007*.
- [8] Zhengyou Zhang. "Flexible Camera Calibration By Viewing a Plane From Unknown Orientations", <http://research.microsoft.com/~zhang>.
- [9] J. Shi and C. Tomasi, "Good Features to Track", *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1994.
- [10] R. Haralick et al., "Review and Analysis of Solutions of the Three Point Perspective Pose Estimation Problem", *International Journal of Computer Vision*, 1994.
- [11] Q. Zhang and R. Pless. "Extrinsic Calibration of Camera and Laser Range Finder", *Proceedings Of 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems, September 28 -October 2, 2004, Sendai, Japan*.
- [12] S. Wasielewski, O. Strauss "Calibration of a multi-sensor system laser rangefinder / camera".
- [13] E. Primo, "Tesi di laurea", *Università degli studi di Catania, Facoltà di Ingegneria* 2008.