



UNIVERSITÀ DEGLI STUDI DI PARMA

FACOLTÀ DI INGEGNERIA

CORSO DI LAUREA SPECIALISTICA IN INGEGNERIA INFORMATICA

REALIZZAZIONE DI UN SISTEMA DI  
ACQUISIZIONE ED ELABORAZIONE DI IMMAGINI  
BASATO SU TELECAMERA DI PROFONDITÀ  
DESIGN OF A RANGE CAMERA-BASED IMAGE  
ACQUISITION AND PROCESSING SYSTEM

Relatore: Chiar.mo Prof. S. CASELLI

Correlatori: Dott. Ing. D. LODI RIZZINI

Dott. Ing. J. ALEOTTI

Tesi di laurea di: LUIGI GALATI

ANNO ACCADEMICO 2010-2011

*Alla mia famiglia*

*A Sara*

# Ringraziamenti

Desidero ringraziare il Prof. Stefano Caselli, per la costante disponibilità e per il tempo dedicatomi durante tutto il periodo di tesi.

Inoltre un ringraziamento va al Prof. Dario Lodi Rizzini, per essere sempre stato presente e disponibile per la risoluzione dei numerosi problemi che si sono verificati durante le varie fasi del mio lavoro.

Un altro ringraziamento va al Prof. Jacopo Aleotti per essere stato sempre a disposizione, soprattutto nella fase più difficile del lavoro.

Un ringraziamento speciale va a tutta la mia famiglia per avermi sempre sostenuto e amato. Perchè solo chi ti ama è disposto ad affrontare i numerosi sacrifici che si sono resi necessari per arrivare a scrivere questa tesi.

Vorrei inoltre ringraziare Sara, che ha affrontato al mio fianco tutti i momenti di questo periodo universitario, siano essi felici o tristi. Un ringraziamento va anche a tutta la famiglia di Sara, che mi ha sempre accolto a braccia aperte. In particolare vorrei ringraziare Fausto, per l'aiuto nella costruzione dell'apparato sperimentale usato in questo lavoro di tesi. Infine, desidero ringraziare tutti i miei amici, per i momenti felici passati insieme.

# Indice

<b>Introduzione</b>	<b>1</b>
<b>1 Stato dell'arte</b>	<b>4</b>
1.1 Introduzione . . . . .	4
1.2 Sistemi di acquisizione basati su laser . . . . .	4
1.2.1 Sensori basati sulla triangolazione . . . . .	5
1.2.2 Sensori basati sul tempo di volo o modulazione di fase . . . . .	7
1.2.3 Attuazione nei sensori basati su laser . . . . .	8
1.3 Sensori basati su telecamera . . . . .	11
1.3.1 Sensori basati su visione stereo . . . . .	11
1.3.2 Sensori basati sul tempo di volo . . . . .	13
1.3.3 Sensori basati su luce strutturata . . . . .	14
<b>2 Principio di funzionamento e studio geometrico</b>	<b>16</b>
2.1 Principio di funzionamento . . . . .	16
2.1.1 Misurazioni con la telecamera . . . . .	17
2.1.2 Misurazioni con encoder . . . . .	18
2.1.3 Componente di Range . . . . .	19
2.2 Principali configurazioni geometriche . . . . .	21
2.3 Sistema di riferimento . . . . .	23

---

2.4	Riflessione . . . . .	24
2.5	Occlusione . . . . .	25
2.6	Angolo di vista e risoluzione . . . . .	25
2.7	Calcolo del Field Of View . . . . .	26
2.8	Risoluzione . . . . .	28
2.9	Costruzione del prototipo del sistema di acquisizione . . . . .	30
2.9.1	Esempio di costruzione del sistema di acquisizione . . . . .	32
<b>3</b>	<b>Realizzazione di un sistema di calibrazione</b>	<b>34</b>
3.1	Introduzione . . . . .	34
3.2	Distorsione . . . . .	35
3.3	Calibro . . . . .	36
3.4	Calibro realizzato in laboratorio . . . . .	38
3.5	Utilizzo del software di calibrazione . . . . .	39
3.5.1	Impostazione dei parametri . . . . .	39
3.5.2	Impostazione della ROI . . . . .	40
3.5.3	Calibrazione . . . . .	41
3.6	L'algoritmo usato in calibrazione . . . . .	45
3.6.1	Correzione della distorsione della lente . . . . .	46
3.6.2	Omografia . . . . .	47
3.6.3	Approcci alternativi . . . . .	48
3.6.4	Coordinate mondo nell'uso senza encoder . . . . .	50
<b>4</b>	<b>Apparato sperimentale e sistema di acquisizione</b>	<b>53</b>
4.1	Introduzione . . . . .	53
4.2	Apparato sperimentale e sua analisi . . . . .	54
4.2.1	Analisi dell'apparato sperimentale . . . . .	55
4.3	Il programma "Ranger Studio" . . . . .	57

---

4.4	Parametri della telecamera:modalità “measurement” . . . . .	58
4.4.1	High-resolution 3D (DCM) . . . . .	61
4.5	Icon API . . . . .	62
4.6	Classe di acquisizione . . . . .	63
4.7	Acquisizione dati . . . . .	66
4.7.1	Icon Buffers, Scans, Profiles e Data Format . . . . .	66
4.7.2	Polling e Call-back . . . . .	68
4.7.3	Filtro di calibrazione . . . . .	69
4.8	Misure . . . . .	69
<b>5</b>	<b>Elaborazione di immagini di profondità</b>	<b>73</b>
5.1	Introduzione . . . . .	73
5.2	Strutture dati per immagini di range . . . . .	74
5.3	Le caratteristiche NARF . . . . .	75
5.4	Clusterizzazione e selezione dei punti caratteristici . . . . .	80
5.4.1	Estrazione del cluster . . . . .	80
5.4.2	Selezione dei punti caratteristici . . . . .	81
5.5	Riconoscimento di oggetti . . . . .	82
5.5.1	Allineamento . . . . .	84
5.5.2	Classificazione in base all’errore di allineamento . . . . .	86
5.5.3	Classificazione in base alla percentuale di matching . . . . .	86
5.6	Risultati . . . . .	87
	<b>Conclusioni</b>	<b>89</b>
	<b>Bibliografia</b>	<b>92</b>

# Introduzione

Nell'ultimo decennio si è assistito alla diffusione di sensori che, pur sfruttando differenti tecnologie, consentono di acquisire informazioni tridimensionali. Le applicazioni possibili sono innumerevoli e vanno dal controllo della qualità sofisticato all'intrattenimento, dall'automazione alla ricostruzione della scena.

Tali sistemi di acquisizione sfruttano tecnologie differenti (laser o visione) ed operano secondo principi di funzionamento diversi; essi sono la triangolazione, la misura del tempo di volo o altre proprietà fisico-geometriche. Spesso i dispositivi sono il risultato dell'applicazione di diversi principi in combinazione tra loro.

Questo lavoro di tesi ha come obiettivo quello di descrivere ed utilizzare un particolare sensore, la telecamera di profondità *Sick Ranger D50*. Questo sensore opera secondo il principio della luce strutturata ed ha come caratteristica principale l'elevata configurabilità. Per poter operare con la *Ranger D50* è necessario che la telecamera sia collocata su un appropriato supporto e solidale con essa un illuminatore laser (con lunghezza d'onda nel vicino infrarosso). La percezione avviene mediante un movimento relativo tra l'insieme telecamera e laser e la scena osservata. Normalmente tale movimento viene realizzato mantenendo fermi la telecamera e l'illuminatore ed utilizzando un tappeto mobile per la movimentazione dell'oggetto che si desidera osservare. Si tratta pertanto di un sensore progettato per applicazioni in impianti industriali. La telecamera *Ranger D50* si contraddistingue per l'elevata risoluzione dell'immagine di profondità ( $1536 \times 512$  pixel) e per l'elevata precisione nelle misurazioni di distanza,

che ha portato in questo lavoro di tesi a rilevare un errore statistico di circa 0,5 mm. Poichè il sensore richiede la presenza di una struttura di supporto, da progettare *ad hoc* per ogni situazione applicativa, la casa costruttrice ha messo a disposizione un metodo per la calibrazione che deve essere eseguito in modo molto accurato per ottenere le migliori prestazioni dal sistema.

L'attività di tesi si è articolata sostanzialmente in due fasi. In primo luogo è stato messo a punto l'apparato sperimentale per i diversi elementi costitutivi del sistema di acquisizione. È stato inoltre necessario realizzare lo strumento di calibrazione (con pattern a dente di sega) e comprendere l'utilizzo del metodo di calibrazione. Successivamente è stato predisposto un programma per l'acquisizione delle immagini usando le API fornite dalla casa costruttrice. La principale difficoltà riscontrata è stata la mancanza di un encoder per la misura dell'avanzamento dell'oggetto rispetto alla telecamera ed all'illuminatore laser. La mancanza di un dato preciso, nella direzione di traslazione dell'oggetto osservato, deve essere sopperita da un'opportuna scelta dell'intervallo temporale di campionamento e da un accurato calcolo dell'immagine di prossimità.

Nella seconda fase è stata sviluppata un'applicazione per il riconoscimento e la classificazione degli oggetti acquisiti con le immagini di range. La classificazione presuppone, in generale, che sia la forma complessiva dell'oggetto sia i dettagli siano ben riconoscibili. Questa considerazione è ancora più valida se il riconoscimento viene effettuato a seguito dell'estrazione di punti caratteristici o feature. I metodi basati su estrazione di caratteristiche, resi popolari dalla diffusione di algoritmi come il SIFT (Scale Invariant Feature Transform) [8], hanno come risultato un insieme di punti cui sono associati dei valori, solitamente denominati descrittori, che dipendono dalle caratteristiche locali e globali di un intorno dell'immagine. Sebbene la letteratura si sia occupata prevalentemente di feature per immagini a toni di grigio o a colori, recentemente sono state introdotte delle caratteristiche che riescono a descrivere le im-

magini di prossimità. Queste caratteristiche sono conosciute con l'acronimo NARF (Normal Aligned Radial Feature) e sono descritte in [17]. Le caratteristiche NARF fanno parte di un progetto più ampio, la libreria open source PCL (Point Cloud Library), che permette la gestione e l'elaborazione di un insieme di punti acquisiti con sensori di profondità.

Tale libreria è disponibile al sito internet <http://www.pointclouds.org>. L'applicazione di classificazione costruita tiene conto delle caratteristiche dell'immagine e di alcune operazioni basilari compiute su di essa. I risultati di classificazione sono risultati soddisfacenti, in particolare per il fatto che le caratteristiche estratte dall'immagine di prossimità sono stabili. La stabilità delle caratteristiche è dovuta soprattutto alla qualità dell'immagine di range, quindi si è riusciti a dimostrare che grazie all'alta risoluzione della telecamera si hanno a disposizione immagini di prossimità di ottima fattura.

La tesi è organizzata come segue:

Il capitolo 1 illustra le varie tipologie di dispositivi per l'acquisizione di informazioni tridimensionali ed i loro principi di funzionamento. Il capitolo 2 descrive nel dettaglio il principio di funzionamento della telecamera Sick Ranger D50 e la sua caratterizzazione geometrica. Nel capitolo 3 si approfondisce il tema della calibrazione della telecamera di profondità e della costruzione del calibro, descrivendo l'algoritmo di calibrazione in dettaglio. Il capitolo 4 illustra il programma elaborato per l'acquisizione delle immagini di prossimità dal sensore. Inoltre introduce i risultati delle misure che sono state effettuate per definire la precisione e l'accuratezza del sensore. Infine, il capitolo 5 descrive le elaborazioni eseguite sulle immagini di range e gli algoritmi utilizzati per sviluppare il classificatore di oggetti con i relativi risultati.

# Capitolo 1

## Stato dell'arte

### 1.1 Introduzione

In questo capitolo si vuole fare una classificazione dei vari tipi di sensori per l'acquisizione di dati di prossimità e descriverne le caratteristiche e le modalità di utilizzo. Per la classificazione verrà preso come riferimento un capitolo sui sensori di range di [5]. La classificazione dei dispositivi può essere fatta sulla base di diverse caratteristiche, per esempio in base alla strutturazione dei dati di uscita, in base alla presenza o meno di attuazione, oppure in base al tipo di attuazione. La distinzione principale effettuata in questo capitolo è quella che tiene conto del tipo di sensore principale presente nel sistema di acquisizione dati, ossia se il sistema di acquisizione è dotato di un sensore principale di tipo laser oppure su di un sensore di visione come la telecamera.

### 1.2 Sistemi di acquisizione basati su laser

Il laser è un dispositivo in grado di emettere un fascio di luce coerente, monocromatica e, con alcune eccezioni, concentrata in un raggio rettilineo estremamente collimato attraverso il processo di emissione stimolata. Inoltre la luminosità delle sorgenti laser è

molto elevata paragonata a quella delle sorgenti luminose tradizionali. Queste tre proprietà (coerenza, monocromaticità e alta luminosità) sono alla base del vasto ventaglio di applicazioni che i dispositivi laser hanno avuto e continuano ad avere nei campi più disparati, tra i quali anche l'acquisizione di dati di prossimità.

Un sistema di acquisizione di dati di range basato su laser necessita inevitabilmente di un sistema di attuazione per riuscire a determinare il dato di profondità.

Oltre ai motivi citati in precedenza i laser sono di larga diffusione anche per i seguenti motivi:

- riescono a generare un fascio luminoso da una sorgente di luce meno luminosa;
- riescono a dare dei fasci di luce stretti;
- una sorgente a singola frequenza permette l'eliminazione più semplice delle frequenze indesiderate.

I dispositivi di acquisizione basati su laser possono essere suddivisi in due categorie:

- sensori basati sulla triangolazione;
- sensori basati sul tempo di volo o modulazione di fase.

### **1.2.1 Sensori basati sulla triangolazione**

Il principio di funzionamento di un sistema di acquisizione basato su triangolazione laser è mostrato in figura 1.1. Il fascio dell'illuminatore laser viene proiettato sull'oggetto da osservare da una determinata posizione, mentre un sensore rileva la componente riflessa. Conoscendo le posizioni relative e gli orientamenti del sensore e del laser insieme al modello geometrico si riesce a determinare la posizione in coordinate tridimensionali del punto illuminato.

Il processo di triangolazione è tanto più accurato quanto più è precisa l'informazione

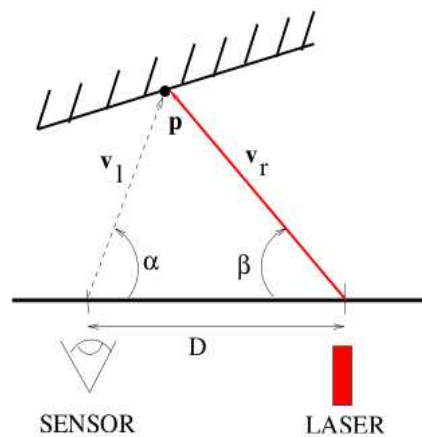


Figura 1.1: Principio di funzionamento della triangolazione laser. Figura presa da [5]

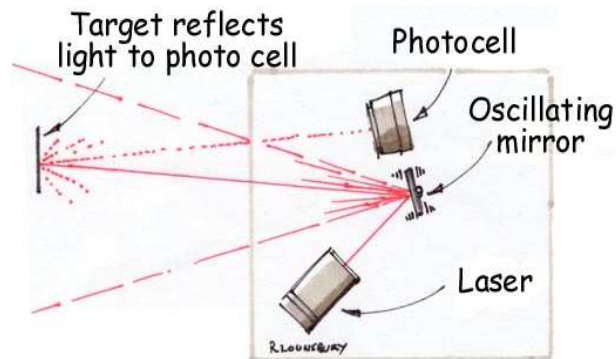


Figura 1.2: Principi di funzionamento del laser scanner

sulla posizione dell'illuminatore laser rispetto al sensore.

L'illuminazione del laser si può controllare, questo comporta dei vantaggi:

- un laser ad una frequenza nota può essere comparato con un filtro spettrale alla stessa frequenza per riconoscere la luce del laser in presenza di altre sorgenti luminose, in modo tale da far comparire nell'immagine la sola luce del laser;
- il fascio laser può essere riflesso da una lente o uno specchio per creare un'insieme di fasci laser che permettono di fare misurazioni di più punti nel piano o nello spazio (come per esempio accade nel *laser scanner*, figura 1.2)



Figura 1.3: Un esempio di laser scanner: *Sick LMS100*

- i raggi laser possono essere orientati, in modo controllato, da specchi. Questo può essere necessario per concentrare il fascio del laser in una determinata area, illuminare un determinato oggetto oppure un determinato ostacolo.

Ci sono anche alcuni svantaggi:

- ci sono potenziali rischi per l'occhio umano, dipendenti dalla classe del laser utilizzato;
- vi è una dipendenza delle prestazioni dal tipo di materiale illuminato. Le riflessioni speculari di oggetti metallici, per esempio, possono creare un disturbo e degradare le prestazioni del sistema.

### 1.2.2 Sensori basati sul tempo di volo o modulazione di fase

I sensori basati sul tempo di volo calcolano la distanza fra la sorgente e la superficie che si desidera misurare calcolando il tempo che la sorgente luminosa impiega per arrivare sulla superficie e tornare sul sensore. Sono basati sulla metodologia impiegata nei sensori radar ma utilizzano la luce.

Il tempo di viaggio moltiplicato per la velocità della luce, dà la distanza dalla superficie misurata.

Il limite per l'accuratezza di questi sensori è rappresentato dal minimo tempo di volo

misurabile e quindi dalla minima distanza misurabile. Molti di questi sensori utilizzano per le misure un intervallo di ambiguità (per esempio 20 metri). Il sensore emette degli impulsi di luce periodicamente e calcola una distanza media dalla superficie da misurare in base al tempo in cui gli ritornano gli impulsi inviati. Per limitare il rumore prodotto dalle riflessioni e semplificare l'elettronica necessaria a ricevere il segnale di ritorno, la maggior parte dei sensori accetta solo segnali che arrivano entro un tempo  $\Delta t$ ; però questa finestra temporale potrebbe osservare anche dei segnali precedenti che vengono riflessi da una superficie più lontana di quella da misurare.

La variazione di fase è un altro principio che è possibile sfruttare per misurare distanze. Se il segnale emesso nell'ambiente è modulato, il raggio riflesso ha un valore di fase differente da quello originario. Tale differenza può essere rilevata sfruttando l'interferenza con il raggio originario. Il valore della variazione di fase è proporzionale al cammino percorso. Di solito i sensori basati su tempo di volo inviano un singolo fascio laser ed è quindi possibile misurare solo un punto alla volta appartenente alla superficie; le applicazioni però richiedono nella maggior parte dei casi, più informazioni, così ora la maggior parte dei sensori restituisce un vettore di informazioni di range, informazioni relative a punti che giacciono su di un piano. Un esempio di sensore laser basato sul tempo di volo è il laser scanner (figura 1.3).

### 1.2.3 Attuazione nei sensori basati su laser

Come si è descritto ed anticipato in precedenza il sensore laser ha bisogno di un sistema di attuazione per misurare distanze nelle diverse direzioni. L'attuazione può anche essere in parte interna al sensore (per esempio lo specchio rotante del laser scanner). Vi sono diversi tipi di sistemi di attuazione che permettono di ottenere delle immagini non strutturate ed altri che se usati in un certo modo permettono di ottenere delle vere e proprie immagini di prossimità.

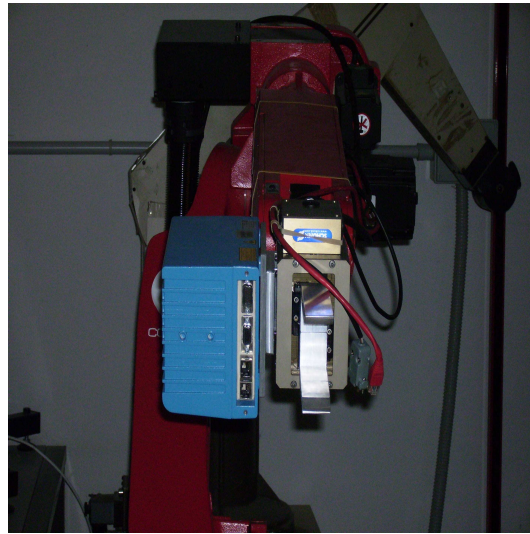


Figura 1.4: Manipolatore *Comau Smart Six* con laser scanner *Sick LMS400*

Per esempio un tipo di attuazione si è già descritto brevemente nel sotto-paragrafo 1.2.1, ossia lo specchio o la lente rotante del laser scanner è un esempio di attuatore. I dati ottenuti dal laser scanner sono strutturati ma di tipo planare. Un laser scanner può anche essere montato su di un manipolatore; in questo caso si possono avere dati non strutturati ma se il manipolatore viene fatto muovere in modo vincolato allora le misure di range risultano organizzate. Nel laboratorio di Robotica dell'Università di Parma è presente un sistema costituito da un laser *Sick LMS400* montato su un manipolatore *Comau Smart Six* (figura 1.4). Un altro tipo di attuatore è il *pan-tilt* per telecamere (figura 1.5), ossia su di un dispositivo che altera in modo controllato gli angoli di beccheggio e di imbardata. Scegliendo opportunamente gli angoli è possibile comporre ad intervalli angolari regolari. Un esempio di questo utilizzo del laser scanner è mostrato in [11] in cui un laser scanner è montato su di un *pan-tilt* per il riconoscimento delle forme di oggetti statici in un ambiente dinamico. Un altro esempio di sistema laser attuato è presente in [7]. La posizione del laser *2D CMOS Serie ZS-L* è modificata dal robot *AlphaCU4* per poter acquisire immagini di prossimità (figura 1.6). In questo caso l'orientamento del laser non viene alterato.



Figura 1.5: Esempio di una struttura *Pan-Tilt*



Figura 1.6: Esempio di movimentazione laser utilizzata in [7]

## 1.3 Sensori basati su telecamera

I sensori di visione, in generale, hanno come modello geometrico di riferimento la proiezione prospettica di tipo *pin-hole*. La telecamera prevede la presenza di un apparato ottico che introduce una distorsione. Può essere necessario, quindi, operare una correzione. Il sensore di visione classico, inoltre, non permette di ottenere direttamente informazioni tridimensionali e richiede complesse operazioni (stereoscopia, luce strutturata) per poter essere sfruttata per l'acquisizione tridimensionale. I sensori basati su telecamera sono sostanzialmente di tre tipologie:

- Sensori basati su visione stereo;
- Sensori basati su tempo di volo;
- Sensori a luce strutturata.

### 1.3.1 Sensori basati su visione stereo

Un modo per ottenere delle immagini di prossimità è quello di acquisire dei dati da diversi sensori; la visione stereo utilizza due o più sensori per acquisire delle immagini ed ottenere la posizione dei punti nella scena.

Il principio di funzionamento è la triangolazione: dallo scostamento della proiezione omografica di un punto su due diverse immagini poste a distanza nota (figura 1.7) è possibile ottenere la profondità [6]. La visione stereo risulta più complicata da utilizzare e meno precisa dei sensori basati su laser. Un esempio di sensore stereo è mostrato in figura 1.8. Un'applicazione basata su visione stereo è visibile in [1] dove gli autori utilizzano la visione stereo su robot mobile per il riconoscimento di ambienti esterni.

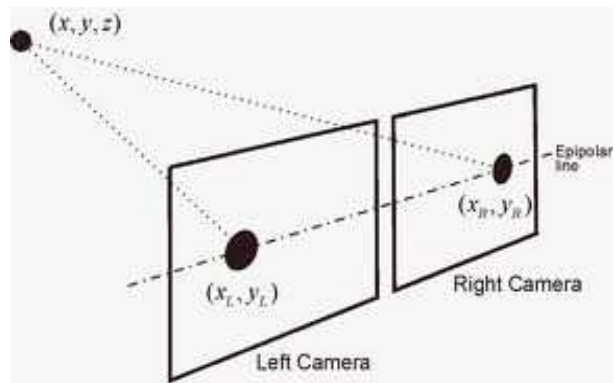


Figura 1.7: Visione stereoscopica

Figura 1.8: Esempio di sensore stereo: il *Bumblebee2*



Figura 1.9: Esempio di telecamera TOF (Time Of Flight): la *Mesa SR4000*

### 1.3.2 Sensori basati sul tempo di volo

Le telecamere a tempo di volo sono in genere caratterizzate da una risoluzione di alcune migliaia di pixel e da un range di misura che varia da alcune decine di centimetri ad un massimo di 30 metri. Queste telecamere sono in grado di fornire in tempo reale l'insieme dei punti  $(x, y, z)$ , l'immagine di prossimità, l'immagine di ampiezza e l'immagine di intensità.

Il principio di misura sul quale si basano le telecamere TOF (Time Of Flight) è quello del calcolo del tempo di volo, ossia l'intervallo trascorso tra l'istante di emissione e quello di ricezione di un segnale. Il principio di funzionamento è simile a quello illustrato nel caso dei laser scanner (figura 1.2). Analogamente si può sfruttare la differenza di fase tra il segnale emesso ed il segnale ricevuto dal sensore (come spiegato prima nel sotto-paragrafo 1.2.2, maggiori dettagli in [3]). La differenza principale rispetto ai sensori laser è la natura del ricevitore. In questo caso il ricevitore è un sensore immagine che raccoglie i dati a matrice. Il range della telecamera in figura va dagli 0,1 metri ai 5 metri; ha un'accuratezza assoluta di circa 10 mm e una deviazione standard su ripetibilità di 4 mm ([9]). Un lavoro in cui si effettua l'acquisizione in tempo reale di immagini di prossimità attraverso una telecamera basata su tempo di volo è visibile in [10].



Figura 1.10: Esempio di telecamera a luce strutturata: *Polhemus FastScan*

### 1.3.3 Sensori basati su luce strutturata

I sensori basati su luce strutturata utilizzano la triangolazione laser per effettuare la misura di profondità, quindi hanno un principio di funzionamento simile a quello descritto nel paragrafo 1.2.1.

Viene utilizzato un fascio laser per illuminare la sezione trasversale alla telecamera dell'oggetto. In questo modo, il sensore, guardando come si riflette la luce del laser sulla matrice, riesce a determinare il dato di prossimità contando le righe che occupa il contorno dell'oggetto delimitato dalla luce del laser. Alcuni esempi di sensori video basati su luce strutturata sono visibili in figura 1.10 ed in figura 1.11. La telecamera in figura 1.10 viene utilizzata soprattutto per la ricostruzione di oggetti in 3D; il principale vantaggio, oltre al fatto di avere un'alta risoluzione, è il fatto di poter essere utilizzata senza un apparato di supporto, ma il movimento viene effettuato direttamente muovendo la telecamera con la mano. La telecamera in figura 1.11 invece ha una risoluzione anch'essa elevata, ma ha bisogno di una struttura di supporto per l'acquisizione delle immagini. Questa telecamera è usata soprattutto per il controllo di qualità degli oggetti nel settore industriale. Queste due telecamere descritte hanno però un costo elevato, dovuto appunto alla alta risoluzione e alla velocità di acquisizione delle immagini. Vi



Figura 1.11: Esempio di telecamera a luce strutturata: *Sick Ranger D50*



Figura 1.12: Il sensore *Kinect*

sono però delle telecamera a luce strutturata a basso costo che vengono utilizzate per lo più per la ricerca accademica o per il settore multimediale. Questo è il caso del *Kinect Microsoft* (figura 1.12), nato inizialmente come controller per una console per videogiochi, ma impiegato oramai per scopi diversi ed anche in lavori di ricerca. Uno di questi è presentato in [18] dove gli autori hanno presentato un lavoro per il riconoscimento di persone utilizzando appunto la telecamera *Kinect*. Il sensore in figura 1.11 verrà trattato in maniera dettagliata in questa tesi, dal capitolo 2 in poi.

## Capitolo 2

# Principio di funzionamento e studio geometrico

In questo capitolo vengono introdotti i principi di funzionamento della telecamera Sick Ranger D50 e descriverne le caratteristiche. Verranno inoltre esaminate le possibili configurazioni geometriche. Dopodichè si presenteranno alcuni problemi, che influiscono sulle prestazioni della telecamera, che derivano dalla geometria del sensore. Per ultimo si andranno ad illustrare le relazioni fra i parametri della telecamera (per esempio FOV, risoluzione, etc.) che concorrono alla determinazione dell'apparato di acquisizione. Nei paragrafi che seguono verranno riassunte molte informazioni estratte dai manuali ufficiali della telecamera [15].

### 2.1 Principio di funzionamento

Il Ranger D50 è una telecamera con tecnologia CMOS, quindi ha una matrice interna che permette al sensore di acquisire immagini ad alta risoluzione (in questo caso  $1536 \times 512$  pixel). La matrice della telecamera ha una dimensione di  $4,864 \times 14,592$  mm ed il pixel ha una dimensione di  $9,5 \times 9,5 \mu\text{m}$ . Il sensore ha una tecnologia a luce

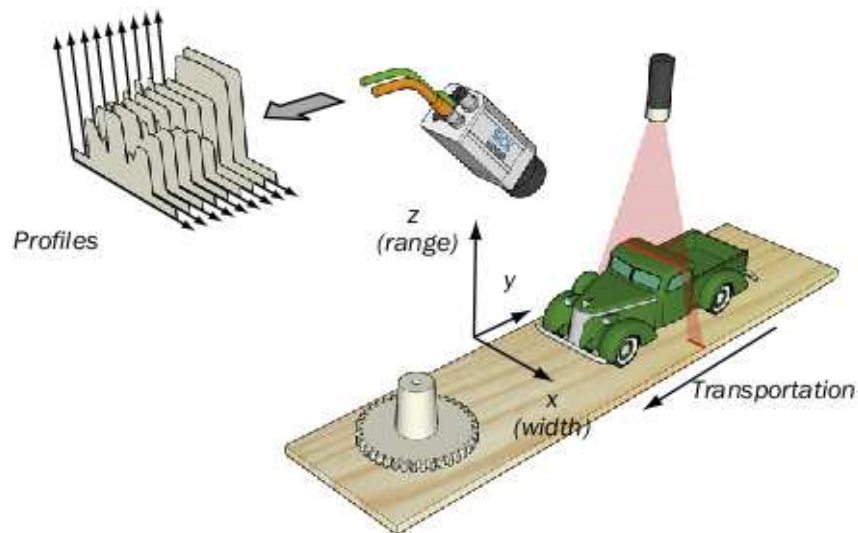


Figura 2.1: Misurazioni tramite la telecamera Ranger D50

strutturata, quindi riesce a determinare il range di un oggetto triangolando con la luce di un fascio laser.

### 2.1.1 Misurazioni con la telecamera

La figura 2.1 presenta uno schema di principio del funzionamento della telecamera. La telecamera esegue le misurazioni sull'asse di movimento  $y$  osservando la proiezione dell'illuminatore laser sull'oggetto lungo l'asse  $x$  e calcolandone l'altezza per tutta la sua larghezza. Per esempio, supponendo di utilizzare la massima risoluzione della telecamera, quindi  $1536 \times 512$  pixel, la telecamera acquisirà 512 profili ("profile"), ed ognuno di essi conterrà 1536 valori che stanno ad indicare l'altezza dell'oggetto ad un determinato  $(x, y)$ . La telecamera costruisce un'immagine di prossimità, in cui la  $x$  e la  $y$  sono rispettivamente gli indici di colonna e di riga, mentre la  $z$  è il valore di range. In figura 2.1 è illustrata una misurazione di un oggetto. Se i valori delle misurazioni della telecamera non sono calibrati allora, per default, sono rappresentati come:

- il valore di range (coordinata  $z$ ) è dato come riga o pixel collocata sul sensore;
- il punto sulla sezione trasversale al movimento (coordinata  $x$ ) è dato come indice di colonna del sensore dove il valore è stato misurato;
- il punto lungo la direzione di movimento (coordinata  $y$ ) è dato come indice di riga del sensore, o come numero sequenziale di scansioni effettuate.

Il sensore può essere impiegato in modalità encoder o in modalità *free*. In modalità encoder le righe dell'immagine di profondità sono acquisite in sincronia con il movimento del tappeto sul quale è posizionato l'oggetto. Ne risulta che le righe dell'immagine ottenuta corrispondono con precisione ad un determinato valore della coordinata di movimento  $y$ . Usando quindi la telecamera con un encoder, si hanno a disposizione le coordinate mondo  $(x, z)$  restituite dalle API della telecamera e la coordinata mondo  $y$  restituita dall'encoder. In modalità *free* le righe dell'immagine sono acquisite ad intervalli temporali costanti. È possibile stimare la coordinata di movimento  $y$  ipotizzando che l'oggetto scorra a velocità costante, per quanto tale stima possa risultare imperfetta e dare luogo a distorsioni. Si consiglia pertanto, se si utilizza la telecamera in modalità *free*, di leggere l'informazione di profondità dalle API e di calcolarsi invece le coordinate  $(x, y)$  come mostrato nel paragrafo 3.6.4.

### 2.1.2 Misurazioni con encoder

Ci sono due modi differenti per usare un segnale esterno per dare un trigger temporale alla telecamera per fare misurazioni (fig. 2.2):

- **Enable:** quando il segnale di enable è alto la telecamera inizia ad eseguire un certo numero di acquisizioni. Dopo queste, se il segnale è ancora alto allora ne esegue delle altre, altrimenti si ferma ed aspetta che il segnale torni alto.
- **Pulse triggering:** indica alla telecamera di eseguire una sola acquisizione.

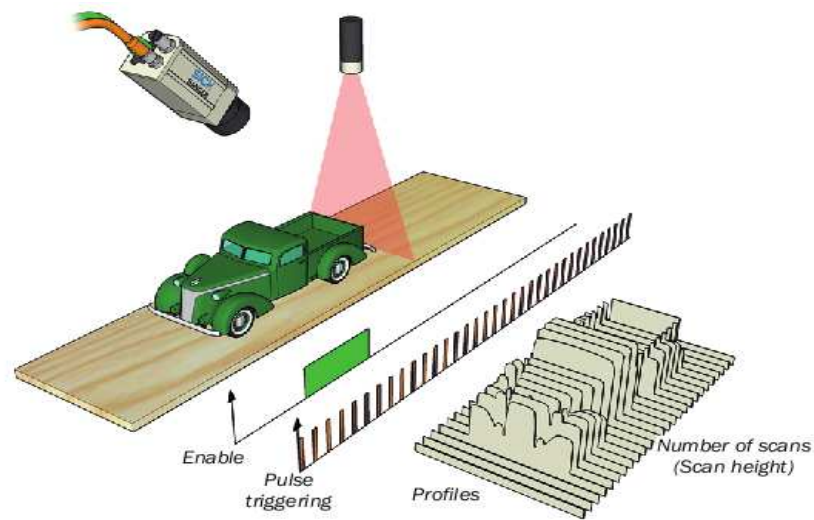


Figura 2.2: Attuazione di misurazioni con un Encoder: mediante *Enable* o *Pulse Triggering*

### 2.1.3 Componente di Range

Una parte importante della configurazione della telecamera è la scelta delle grandezza che si desiderano misurare. Infatti alcune varianti della telecamera *Ranger D50* permettono di misurare, oltre al range, anche l'intensità e le componenti del colore. La componente di range consente le ricostruzioni 3D degli oggetti. La telecamera utilizza la triangolazione laser ed il principio della luce strutturata per effettuare la misura della profondità, ciò significa che l'oggetto viene illuminato dal laser da una direzione, mentre il sensore vede l'oggetto da un'altra direzione. Il laser permette di rendere ben visibile la sezione trasversale dell'oggetto, in modo tale che il Ranger riesce a determinare l'altezza di ogni punto della sezione che si trova sulla verticale della linea del laser (figura 2.3). La tabella 2.1 mostra le componenti che possono essere usate nelle varie range cameras della Sick e si può vedere che la telecamera presente nel laboratorio di robotica dell'Università di Parma, essendo una D, ha solo la componente di range. Per ogni punto misurato la telecamera ritorna il numero di righe, o pixels verticali, dal basso o dall'alto della ROI (Region Of Interest) dove viene trovata la linea del

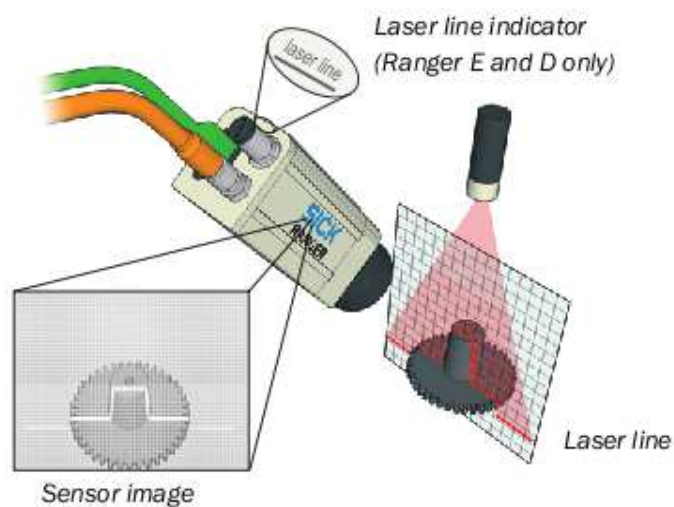


Figura 2.3: Triangolazione laser

laser (figura 2.4). Se la telecamera non riesce a trovare la linea del laser, perchè per

Metodi misurazione range	Modello C	Modello E	Modello D
Horizontal threshold	X	X	-
Horizontal max	X	X	-
Horizontal max and threshold	-	X	-
High-resolution 3D (Hi3D)	-	X	X
High-resolution 3D (Hi3D COG)	X	X	-

Tabella 2.1: Possibili metodi per il calcolo del range nelle varie tipologie di telecamera della Sick

esempio è occlusa oppure è fuori dalla ROI (con ROI si intende l'area che occupa la matrice del sensore, suddivisa in righe e colonne), allora la telecamera ritorna il valore 0. La risoluzione della misurazione dipende dalla componente utilizzata, per esempio se si utilizza la "Horizontal Max" allora si avrà una risoluzione di 1/2 pixel, mentre se si utilizza, come nel caso della telecamera in laboratorio, la componente Hi3D allora si ha una risoluzione di 1/16-imo di pixel. Mentre le prestazioni della telecamera, ossia quanti profili al secondo può misurare, dipendono fortemente dal metodo scelto per effettuare le misurazioni ma anche dall'altezza della ROI scelta. Infatti più righe ha la

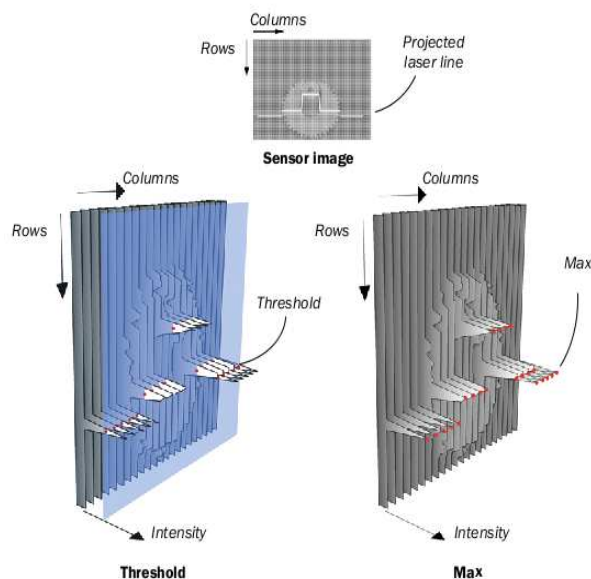


Figura 2.4: Diversi metodi per trovare il range nell’immagine: il metodo *Horizontal threshold* determina il range prendendo l’intensità sopra una certa soglia, mentre *Horizontal max* trova la massima intensità di ogni colonna

ROI più la scansione sarà lenta, mentre diminuendo le righe della ROI si avrà una scansione più veloce. (figura 2.5) La telecamera ed il fascio laser devono essere orientati in modo tale che la linea del laser sia parallela alle righe della matrice del sensore sul Ranger; per questo motivo le telecamere Ranger D ed E, dietro, hanno l’indicazione “laser line” che indica come ci si aspetta che sia orientato il fascio laser.

## 2.2 Principali configurazioni geometriche

Vi sono varie configurazioni geometriche che è possibile adottare attraverso la diversa disposizione della telecamera e del laser fra di loro. Le principali geometrie che si possono adottare sono quattro:

1. **Ordinary:** la telecamera è montata perpendicolarmente al piano di lavoro, mentre il laser, a lato della telecamera forma un angolo  $\beta$  con la normale al pi-

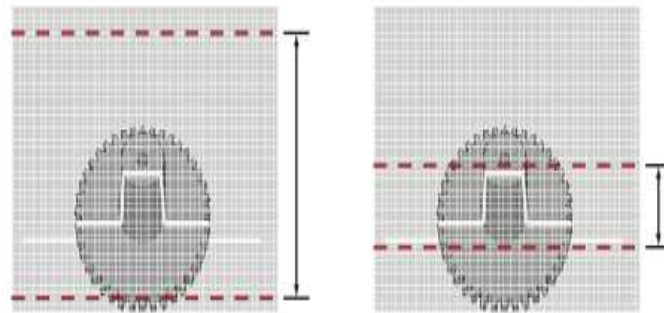


Figura 2.5: Una ROI (delimitata in rosso) con poche righe è più veloce da analizzare in confronto ad una con molte righe

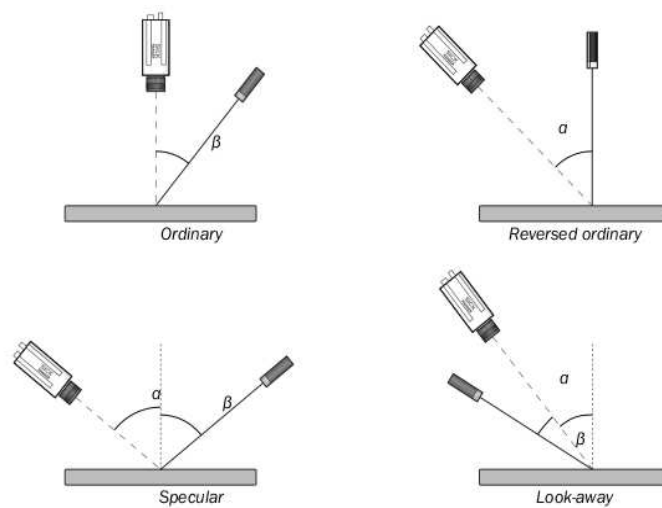


Figura 2.6: Principali configurazioni geometriche

ano di lavoro. Queste geometria è ideale per le telecamere dotate di multiscan (non la RangerD50) ed ha un'alta risoluzione di range. Questa geometria ha lo svantaggio di dipendere dalle riflessioni diffuse del laser.

2. **Reversed Ordinary:** questa geometria è simile alla Ordinary ma le posizioni della telecamera e del laser vengono invertite. Si forma così un angolo  $\alpha$  fra la telecamera e la normale al piano di lavoro. Il principale vantaggio derivante dall'uso di questa geometria è la facile calibrazione, attraverso il software dato in dotazione con la telecamera.
3. **Specular:** in questa geometria laser e telecamera sono montate su lati opposti rispetto alla normale. Adottando questa geometria si ha il vantaggio di poter misurare bene in 2D su superfici lucide, soprattutto per il rilevamento delle crepe. Inoltre si riescono a fare misurazioni in 3D su superfici opache o con poca luce. Infatti questa geometria, rispetto alle altre, necessita di poca illuminazione esterna. Uno svantaggio invece è che se  $\alpha = \beta$  allora il sensore va in saturazione, quindi bisogna fare attenzione ad evitare questa situazione. Inoltre la misurazione 3D su superfici lucide è pessima.
4. **Look-Away:** in questa geometria, telecamera e laser, sono montati sullo stesso lato rispetto alla normale. Si può adottare se si vuole avere una bassa dipendenza da riflessioni, ma si ha una bassa risoluzione delle misure di distanza, molta occlusione e necessita di una forte illuminazione esterna.

## 2.3 Sistema di riferimento

Il sistema di riferimento più appropriato dipende dalla geometria adottata. Nel caso dell'applicazione di questa tesi si è scelto di adottare la Reverse Ordinary. Il sistema di riferimento pertanto è stato definito come segue (figura 2.7):

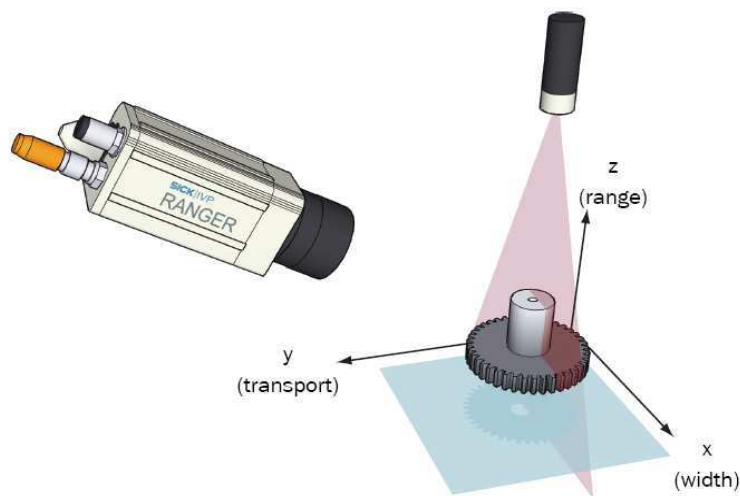


Figura 2.7: Sistema di riferimento per la geometria Reverse Ordinary

- Asse x: ortogonale alla direzione di movimento.
- Asse y: lungo la direzione di movimento.
- Asse z: lungo l'asse ottico del laser.

## 2.4 Riflessione

Quando si fanno delle misurazioni su degli oggetti bisogna tener conto anche delle riflessioni prodotte dalla luce del laser. Un oggetto illuminato può riflettere la luce in diverse direzioni. Nelle superfici lucide la luce viene riflessa con lo stesso angolo rispetto alla normale con cui è arrivata: in questo caso si parla di riflessione diretta o speculare. Le superfici opache, invece, riflettono la luce in più direzioni e con diversi angoli rispetto a quello d'arrivo; in questo caso si parla di riflessione diffusa. Se parte della luce non viene riflessa, può essere assorbita dal materiale, oppure può essere trasmessa attraverso l'oggetto (figura 2.8).

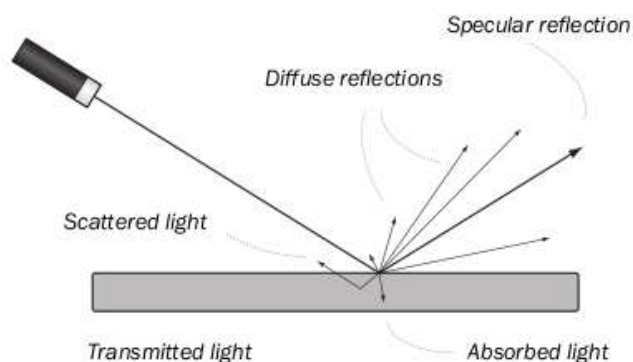


Figura 2.8: Diversi tipi di riflessione del laser

## 2.5 Occlusione

L'occlusione si verifica quando la linea proiettata dall'illuminatore laser non è visibile dalla telecamera. Questo può accadere per due motivi (figura 2.9):

- occlusione del laser: il laser non riesce fisicamente ad illuminare una parte dell'oggetto;
- occlusione della telecamera: quando una parte della linea del laser non è visibile dalla telecamera poichè l'oggetto stesso la occlude.

## 2.6 Angolo di vista e risoluzione

Il range massimo delle misurazioni è il rapporto tra il punto più alto e quello più basso misurabile all'interno di una ROI (Region Of Interest). Avere un range massimo elevato significa avere la possibilità di misurare oggetti che variano anche di molto in altezza. La risoluzione di range è la minima variazione d'altezza che è possibile a misurare. Quindi avere un'alta risoluzione significa che si possono misurare le piccole variazioni d'altezza ma anche che il range massimo misurabile sarà piccolo rispetto a quello di una bassa risoluzione con la stessa ROI. In generale il range massimo e



Figura 2.9: Esempio dei diversi tipi di occlusione

la risoluzione dipendono dall'angolo fra la telecamera ed il laser. Se questo angolo è piccolo si ha che la linea del laser non varia molto nell'immagine del sensore anche se l'oggetto varia molto in altezza; questo si traduce con un elevato range massimo misurabile ma ad una bassa risoluzione, con poca occlusione. Invece se l'angolo è grande allora si avrà che anche piccole variazioni d'altezza spostano la linea del laser anche di diversi pixel all'interno dell'immagine del sensore e questo si traduce con un basso range massimo ma un'alta risoluzione (figura 2.10).

## 2.7 Calcolo del Field Of View

Il calcolo del Field Of View (FOV) è importante per garantire che l'applicazione funzioni bene e per impostare correttamente diversi parametri della telecamera, soprattutto nella fase di calibrazione. Ovviamente il FOV dipende da diversi parametri, alcuni interni del sensore, altri esterni e modificabili dall'utente. La relazione che permette di

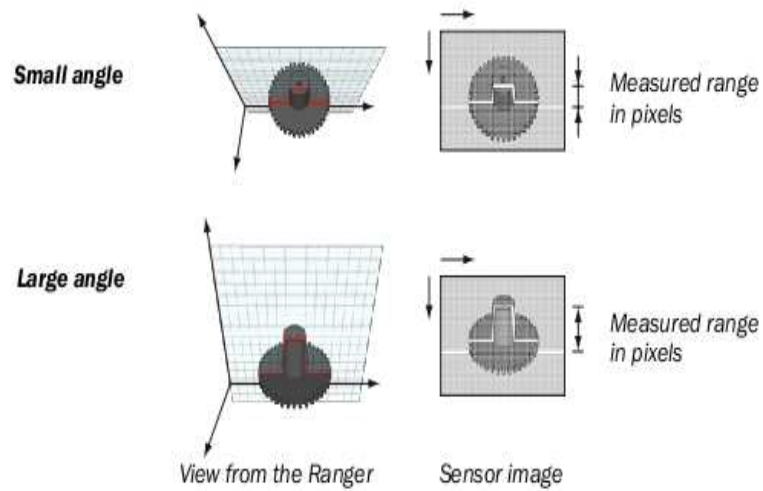


Figura 2.10: Variazione della risoluzione in funzione dell'angolo fra illuminatore laser e telecamera

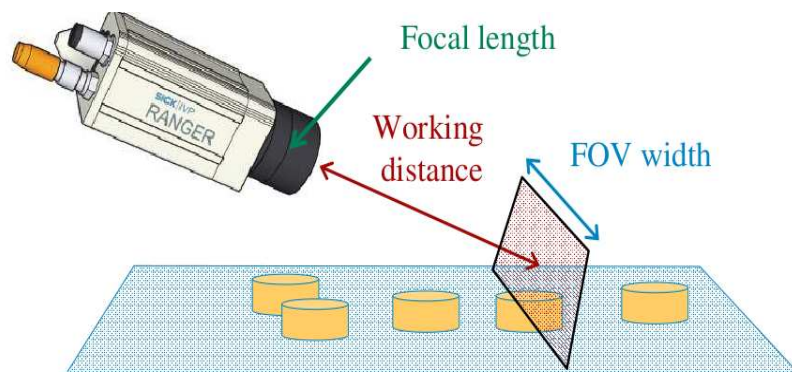


Figura 2.11: Relazioni per il calcolo del Field Of View

calcolare il FOV è la seguente (figura 2.11):

$$\frac{FOV_{width(height)}}{WorkingDistance} = \frac{Sensor_{width(height)}}{FocalLenght}$$

$Sensor_{width}$  costante poichè indica la larghezza fisica del sensore, il parametro  $FocalLenght$  dipende dall'ottica che si sceglie per la telecamera poichè indica la distanza focale, mentre la  $WorkingDistance$  un parametro che si può modificare e scegliere al momento della costruzione dell'apparato di acquisizione. Il parametro  $Sensor_{width}$  vale 14,6 mm (4,9 per il  $Sensor_{height}$ ). In questo progetto si è scelta un'ottica con una  $FocalLenght$  di 25 mm. Il calcolo del FOV quindi diventa:

$$FOV_{width} = \left( \frac{Sensor_{width}}{FocalLenght} \right) \cdot WorkingDistance = \left( \frac{14,6}{25} \right) \cdot WorkingDistance \quad (2.1)$$

## 2.8 Risoluzione

Si può andare ora a calcolare il valore della risoluzione sui vari assi del sistema di riferimento della telecamera. La risoluzione media lungo l'asse x è data dalla seguente formula:

$$X_{res} = \frac{FOV_{width}}{numeroDiPixel} [mm/pixel] \quad (2.2)$$

dove  $FOV_{width}$  è la larghezza del FOV calcolato come indicato nel precedente paragrafo, mentre con  $numeroDiPixel$  si intende il numero dei pixel che ha la massima risoluzione del sensore. In questo caso la telecamera ha una risoluzione di  $1536 \times 512$  pixel (larghezza  $\times$  altezza) quindi visto che stiamo considerando la  $FOV_{width}$  il numero di pixel è 1536. Nel caso in cui si consideri la  $FOV_{height}$  allora il numero di pixel da considerare è 512.

La risoluzione lungo l'asse z dipende dalla geometria laser-telecamera scelta, secondo la seguente tabella 2.2: Le formule in tabella danno la risoluzione per i pixels nel

Geometry	Approximate range resolution
Ordinary	$\Delta Z \approx \Delta X / \tan(\alpha)$
Reversed Ordinary	$\Delta Z \approx \Delta X / \sin(\alpha)$
Specular	$\Delta Z \approx \Delta X \cdot \cos(\beta) / \sin(\alpha + \beta)$
Look-away	$\Delta Z \approx \Delta X \cdot \cos(\alpha + \beta) / \sin(\beta)$

Tabella 2.2: Risoluzione sull'asse z in base alla geometria adottata

loro insieme. Se le misurazioni sono effettuate utilizzando i sub-pixels, allora bisogna dividere questo valore per la risoluzione dei sub-pixels. Per esempio se le misurazioni sono fatte utilizzando la componente Hi3D, che teoricamente ha una risoluzione di 1/16 di pixel allora la risoluzione lungo l'asse z diventa:

$$Z_{res} = \frac{\Delta Z}{16}$$

La risoluzione lungo l'asse y non ha nulla in comune con la risoluzione lungo l'asse x e l'asse z. Può essere vista come la distanza fra i profili. Nel caso in cui le misurazioni vengano effettuate senza l'ausilio di un encoder (quindi in modalità *Free*) allora la risoluzione sull'asse y dipenderà dalla velocità di movimento dell'oggetto e dal *cycle time*, che è un parametro modificabile della telecamera:

$$Y_{res}[mm/scan] = Velocità_{oggetto}[mm/s] \cdot cycleTime[s/scan] \quad (2.3)$$

Nel caso si utilizzi un encoder, allora nel calcolo della risoluzione intervengono i parametri dell'encoder:

$$Y_{res}[mm/scan] = \frac{PulsePerScan}{(EncoderResolution \cdot ScaleFactorOfSignalTrig)} \quad (2.4)$$

*PulsePerScan* è un parametro modificabile della telecamera, mentre gli altri parametri sono caratteristiche dell'encoder.

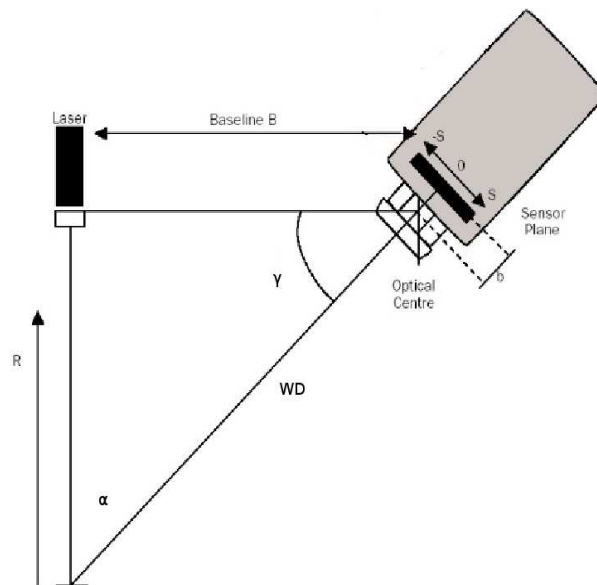


Figura 2.12: Geometria Reversed Ordinary con indicazione dei parametri della Baseline

## 2.9 Costruzione del prototipo del sistema di acquisizione

La costruzione del sistema di acquisizione è sicuramente uno dei punti fondamentali per la buona riuscita dell'applicazione e per avere il funzionamento ottimo della telecamera nelle situazioni di nostro interesse. Infatti non esiste in questo caso un setup ottimale per ogni situazione, essendo la telecamera uno strumento versatile, ma si può calcolare quale sia il setup migliore in base alle esigenze dell'applicazione.

Si vedrà che esistono delle relazioni tra le “variabili” in gioco nella costruzione del setup, cioè tra FOV e risoluzione lungo gli assi, oppure tra working distance e l'angolo dato dalla geometria. Tuttavia le relazioni che si vedranno sono valide solo per la geometria “Reversed Ordinary”; scegliendo questa geometria ci si ritrova nella situazione mostrata in figura 2.12. Supponendo che il centro dell'ottica ed il laser siano perfettamente allineati fra di loro la Baseline B è perfettamente orizzontale e forma un angolo di  $90^\circ$  con la normale al piano di lavoro. In questo modo si ha che la normale, la Baseline B e la Working Distance (WD) formano un triangolo rettangolo. Quindi si

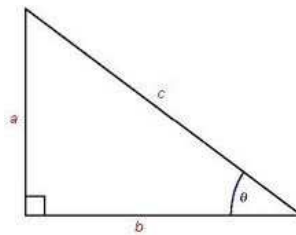


Figura 2.13: Generico triangolo rettangolo

conosce già il valore di uno dei tre angoli, ossia l'angolo retto, mentre l'angolo fra WD e B è chiamato  $\alpha$  e l'altro, fra la normale e la WD è chiamato  $\gamma$ . Essendo un triangolo si deve avere:

$$\alpha + \gamma + 90^\circ = 180^\circ$$

Inoltre prendendo in considerazione un triangolo rettangolo generico come quello in figura 2.13 si hanno le seguenti relazioni:

$$a = c \cdot \sin(\theta) \quad (2.5)$$

$$b = c \cdot \cos(\theta) \quad (2.6)$$

Quindi in questo caso, riprendendo il triangolo in figura 2.12 si verifica:

$$B = WD \cdot \sin(\alpha) \quad (2.7)$$

$$R = WD \cdot \cos(\alpha) \quad (2.8)$$

Così facendo si riesce ad impostare l'angolo fra telecamera e laser in base all'altezza del laser o working distance desiderata, ma non solo, si riescono anche a ricavare le varie risoluzioni lungo gli assi in funzione della WD, della Baseline o dell'altezza del

laser. Unendo la 2.1 alla 2.8 infatti si ha:

$$X_{res} = \left( \frac{Sensor_{width}}{FocalLength \cdot NumeroDiPixel} \right) \cdot \left( \frac{B}{\sin(\alpha)} \right) \quad (2.9)$$

Oppure unendo la 2.1 alla 2.7 si ha:

$$X_{res} = \left( \frac{Sensor_{width}}{FocalLength \cdot NumeroDiPixel} \right) \cdot \left( \frac{R}{\cos(\alpha)} \right) \quad (2.10)$$

Dato che la risoluzione lungo l'asse x è strettamente legata alla risoluzione lungo l'asse z tramite la relazione

$$Z_{res} = \frac{X_{res}}{\sin(\alpha)}$$

allora si riesce ad ottenere anche la risoluzione lungo l'asse z in base alla WD, Baseline, etc...

### 2.9.1 Esempio di costruzione del sistema di acquisizione

Per meglio vedere come usare le precedenti relazioni, si farà un esempio di costruzione di setup. Si supponga di avere le seguenti specifiche:

- baseline di 100 mm
- focal length di 25 mm

Un effetto di cui bisogna tener conto nella costruzione del setup è la distanza minima a cui bisogna avere l'oggetto per metterlo a fuoco (MOD Minimum Object Distance); essa è una caratteristica dell'ottica e nel nostro caso essa è di circa 0,5 metri. Non vi è una formula precisa per calcolarla ma di solito è segnata nel datasheet dell'ottica oppure si può approssimare prendendo come MOD la distanza minima segnata sulla ghiera dell'iride. Supponendo quindi di prendere una working distance (WD)

di 500 mm, è possibile calcolare l'angolo  $\alpha$  che consente di soddisfare le specifiche tramite la 2.8:

$$\alpha = \arcsin\left(\frac{B}{WD}\right) = \arcsin(0,2) = 11,54^\circ \quad (2.11)$$

Determinato l'angolo, si può determinare l'altezza del laser (tramite la 2.7) che deve essere

$$R = WD \cdot \cos(\alpha) = 500 \cdot \cos(0,201) = 489,93 \text{ mm}$$

Quindi nell'esempio considerato il laser deve stare ad un'altezza di circa 490 mm.

Si può ora calcolare il Field Of View (FOV) di questo esempio tramite la 2.1:

$$FOV_{width} = \left(\frac{Sensor_{width}}{FocalLength}\right) \cdot WorkingDistance = \left(\frac{14.6}{25}\right) \cdot 500 = 292 \text{ mm}$$

Quindi ora si hanno tutti i dati necessari per calcolare la risoluzione lungo l'asse x

$$X_{res} = \frac{FOV_{width}}{1536} = \frac{292}{1536} = 0,1901 \text{ [mm/pixel]}$$

e sull'asse z

$$Z_{res} = \frac{X_{res}}{\sin(\alpha)} = \frac{0,1901}{\sin(11,59^\circ)} = 0,952 \text{ [mm/pixel]}$$

La risoluzione lungo l'asse y non può essere calcolata se non conoscendo la velocità di movimento dell'oggetto.

Ad esempio supponendo di avere una velocità  $v = 100[\text{mm/s}]$ , e di usare un cycle time di  $0.003 \text{ [s/scan]}$ . Conoscendo questi dati e supponendo di essere in modalità "free", quindi senza encoder, si può calcolare la risoluzione lungo l'asse y tramite la 2.3:

$$Y_{res} = 100 \cdot 0,003 = 0,3[\text{mm/scan}]$$

## Capitolo 3

# Realizzazione di un sistema di calibrazione

### 3.1 Introduzione

In questo capitolo saranno illustrate la procedura di calibrazione con particolare attenzione al calibro. Dopodichè si spiegheranno le varie fasi di calibrazione, utilizzando lo strumento software *Coordinator* fornito dalla Sick insieme alla telecamera. È necessario prestare molta cura in questa fase, poich solo con una buona calibrazione si riescono ad ottenere prestazioni ottimali dal dispositivo, soprattutto in termini di precisione nella misurazione del range. Lo scopo della calibrazione è quello di determinare i parametri che consentono il calcolo delle distanze a partire dall'immagine a toni di grigio acquisita con il dispositivo.

Con la calibrazione si riesce ad ottenere la trasformazione dalle coordinate del sensore (riga,colonna) in coordinate mondo  $(x,z)$ . La trasformazione dipende da numerosi fattori come la distanza fra la telecamera e l'oggetto, l'angolo fra la telecamera ed il laser e ovviamente le proprietà della lente usata. Il risultato di questa procedura è la costruzione di una look-up table (LUT) i cui valori sono determinati empiricamente. Il

software fornito dalla casa costruttrice permette di effettuare la calibrazione abbastanza velocemente (la velocità dipende anche dalla qualità del calibro realizzato) e semplicemente. La maggiore limitazione riguarda la validità della calibrazione standard: essa è valida esclusivamente con la geometria “Reversed Ordinary”. Se si volesse utilizzare la telecamera in un’altra configurazione geometrica, occorrerebbe ridefinire una procedura “ad hoc” per la sua configurazione prescelta. Il software *Coordinator* utilizza un metodo globale; questo significa che l’oggetto è catturato in una parte del Field of view ma gli effetti e il ri-calcolo dell’errore vengono determinati sull’intera immagine. Nella fase di calibrazione bisogna prestare molta attenzione all’ampiezza del fascio laser: esso deve essere molto sottile in modo tale da rimanere ben all’interno del calibro ed in modo tale che si riesca a riflettere al meglio il fascio. Un’altra accortezza deve essere quella della regolazione dell’apertura dell’iride, poichè se l’ottica cattura troppa luce il fascio laser nella fase di calibrazione risulterà molto rumoroso (problema visibile a video).

## 3.2 Distorsione

Nella telecamera, le proprietà della lente creano delle distorsioni nelle immagini. Le parti centrali delle immagini risultano espanse mentre le parti periferiche risultano ristrette. Questa distorsione viene chiamata distorsione di lente.

Un altro tipo di distorsione, invece, fa apparire l’oggetto più grande o più piccolo in base alla distanza dalla telecamera, ossia la distorsione di prospettiva.

Nell’immagine 3.1 sono mostrati gli effetti sull’immagine delle due distorsioni e quello che si ottiene unendole.

Utilizzando il software “Coordinator” si riesce ad eliminare entrambe le distorsioni ottenendo un’immagine che rispecchia il mondo reale.

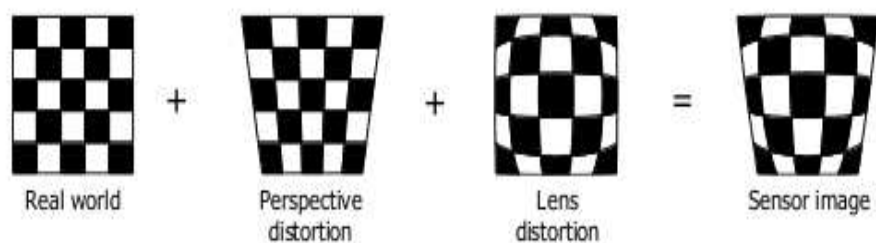


Figura 3.1: Effetti della distorsione prospettica e di lente

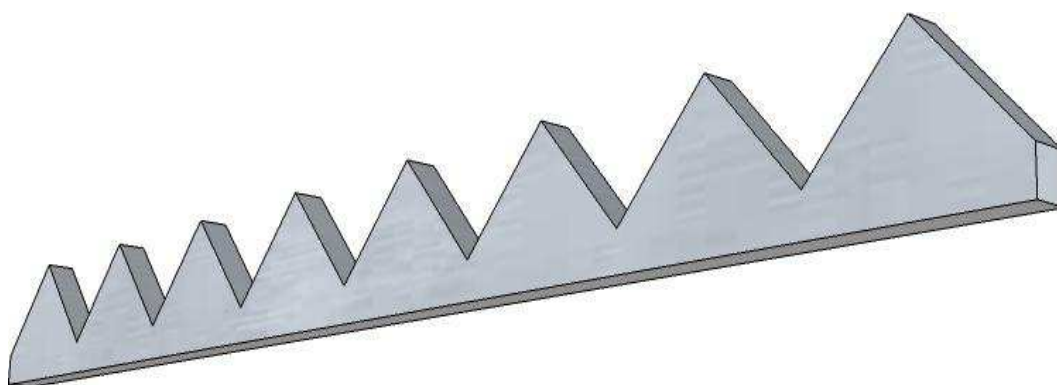


Figura 3.2: Strumento di calibrazione

### 3.3 Calibro

Lo strumento di calibrazione, come già accennato, rappresenta la parte fondamentale di tutto il processo di calibrazione, *poiché* un calibro mal fatto può dare origine a riflessioni del fascio laser che disturbano la buona riuscita della calibrazione; quindi bisogna rispettare scrupolosamente tutte le specifiche di seguito riportate. Come mostrato in figura 3.2 lo strumento di calibrazione si presenta con due lati: un lato è piatto ed è la base dello strumento e l'altro presenta un profilo a dente di sega. Nella fase di calibrazione vengono utilizzati entrambi i lati, quindi bisogna prestare attenzione alla realizzazione degli stessi.

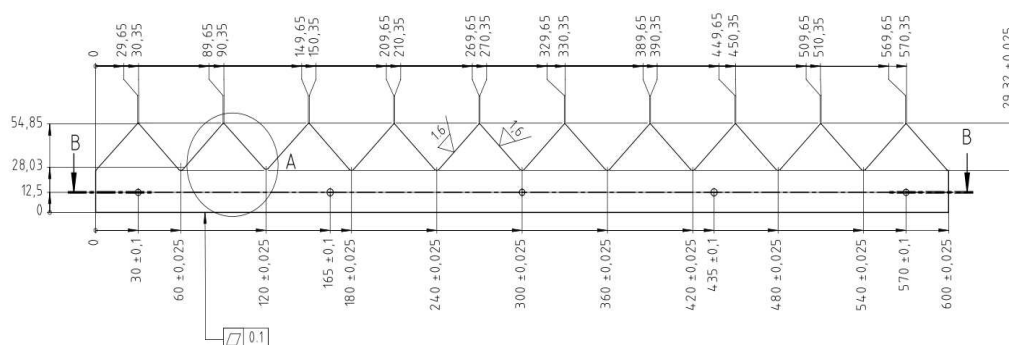


Figura 3.3: Disegno tecnico del calibro

I requisiti minimi che il calibro deve soddisfare e sono di seguito elencati:

- Lo strumento di calibrazione deve coprire almeno il 50% della larghezza della field of view, ma è raccomandata la copertura di almeno il 70%;
- Il field of view deve ricoprire almeno 4 denti del calibro, coprendo sia i picchi che le valli;
- L'angolo fra i picchi e le valli dei denti deve essere maggiore o uguale a  $90^\circ$ , questo significa che la larghezza del dente deve essere maggiore o uguale del doppio dell'altezza del dente ( $larghezzaDente \geq 2 \times altezzaDente$ ).

È consigliata la realizzazione del calibro in alluminio (ma potrebbe andare bene anche un altro materiale che rifletta bene il fascio laser), lo stesso deve essere anodizzato in grigio naturale senza colore. Lo spessore dell'alluminio deve essere di 8 mm. Il disegno tecnico del calibro è mostrato in figura 3.3:

È da notare che il numero dei punti campione necessari per eseguire la calibrazione varia a seconda della larghezza del field of view scelto. Per esempio con la telecamera Ranger D50 quando si calibra utilizzando 1536 pixel bisogna che ogni dente sia largo almeno 50 pixels; Utilizzando metà pixels, quindi 758 pixels, invece, ogni dente deve

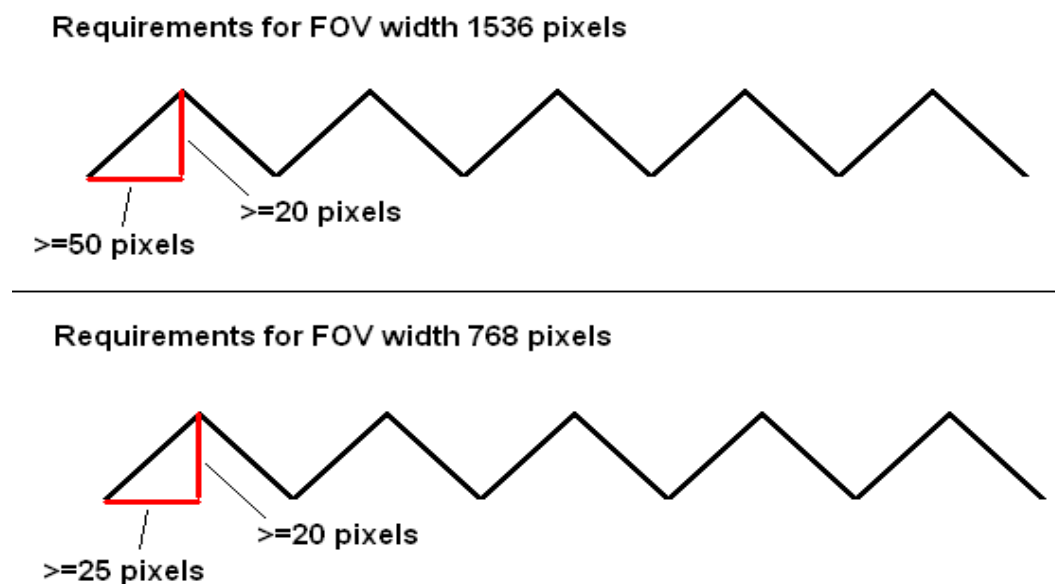


Figura 3.4: Misure del dente di sega in base alla larghezza FOV utilizzata

avere almeno 25 pixel. L'altezza del dente invece non varia in base ai pixel utilizzati, essa deve avere almeno 20 pixel.

Si raccomanda comunque, se si utilizza l'intera dimensione del sensore (1536 pixels), un numero di denti dai 6 agli 8 per la realizzazione del calibro.

### 3.4 Calibro realizzato in laboratorio

Per effettuare la calibrazione sulla telecamera disponibile in laboratorio, si è dovuto costruire il calibro rispettando le specifiche illustrate in precedenza. Le specifiche del calibro usato in laboratorio (prodotto presso il laboratorio di Fisica dell'Università di Parma, che si ringrazia) e visibile nell'immagine 3.5 sono le seguenti:

- **Materiale:** alluminio anodizzato grigio naturale senza colore;
- **Numero di denti:** 8;



Figura 3.5: Strumento di calibrazione utilizzato in laboratorio

- **Spessore:** 8 mm;
- **Larghezza singolo dente:** 48,5 mm;
- **Altezza del singolo dente:** 25 mm;
- **Angolo fra valle e picco:**  $\sim 90^\circ$ .

## 3.5 Utilizzo del software di calibrazione

### 3.5.1 Impostazione dei parametri

Il primo passo nell'utilizzo del software di calibrazione è la scelta della telecamera e del file dei parametri da utilizzare; bisogna inserire l'indirizzo IP della telecamera e nel caso si voglia si può scegliere un file dei parametri dal proprio disco (figura 3.6).

Confermando le scelte fatte si passa al secondo step dove bisogna inserire le specifiche del calibro, ossia l'altezza e la larghezza di un singolo dente di sega.

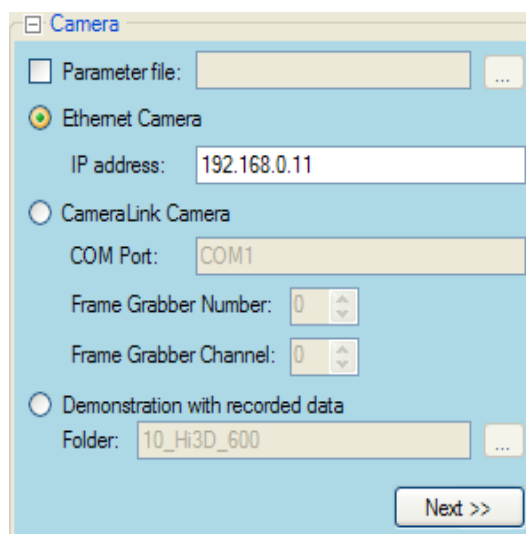


Figura 3.6: Scelta della telecamera e del file dei parametri

### 3.5.2 Impostazione della ROI

Nel terzo step va scelta la dimensione della ROI su cui effettuare la calibrazione (figura 3.7). Di default la ROI viene dimensionata in base alle specifiche settate nel file dei parametri; se in questo passo vengono fatte delle scelte diverse da quelle del file, sullo schermo appare in giallo la ROI modificata mentre in bianco viene disegnata quella impostata nel file dei parametri. Per impostare la ROI bisogna scegliere la riga e la colonna di inizio e il numero stesso di colonne e righe. All'interno della ROI vi è una griglia il cui numero di celle è modificabile dall'utente. È importante che il dente di sega possa essere visualizzato all'interno di una cella.

Per aumentare l'accuratezza del risultato è consigliabile utilizzare il numero più elevato possibile di righe e colonne per formare la griglia. Ovviamente la calibrazione in presenza di un numero di celle elevato richiede più tempo, quindi è opportuno trovare il giusto compromesso in base alle esigenze della propria applicazione. Scelta la ROI si può passare alla vera e propria calibrazione.



Figura 3.7: Scelta della ROI e della griglia

### 3.5.3 Calibrazione

Vi sono due fasi di calibrazione: una in cui è utilizzato il lato piatto ed una in cui viene utilizzato il lato a dente di sega del calibro. Per iniziare la calibrazione si deve premere il tasto “Start”. La calibrazione si può in ogni momento sospendere premendo il tasto “Pause”. Si inizia a calibrare utilizzando il lato piatto del calibro.

Come si può vedere in figura 3.8 vi sono due finestre: quella più in alto è utilizzata per questa fase, vi è visualizzata la ROI con le varie celle. In ogni cella verrà visualizzato l’errore che si ha all’interno della stessa, ovviamente minore è l’errore e migliore è la calibrazione. Si può impostare un errore massimo ammissibile, in modo tale che la visualizzazione della ROI venga facilitata in base ai colori, verde per le celle che presentano un errore più basso di quello impostato e rosso per le celle che lo sfiorano (risultato visualizzabile in figura 3.9). Per fare questo bisogna spuntare la casella “Error Threshold” e impostare la soglia d’errore desiderata.

Il lato piatto del calibro va tenuto fermo all’interno della ROI finchè la telecamera non acquisisce un profilo completo. La completa acquisizione di un profilo è annunciata con un segnale acustico (che si può disabilitare).

Una volta acquisito un profilo si deve ruotare il calibro lungo l’asse orizzontale (come mostrato in figura 3.11), spaziando per tutta la ROI e facendo attenzione

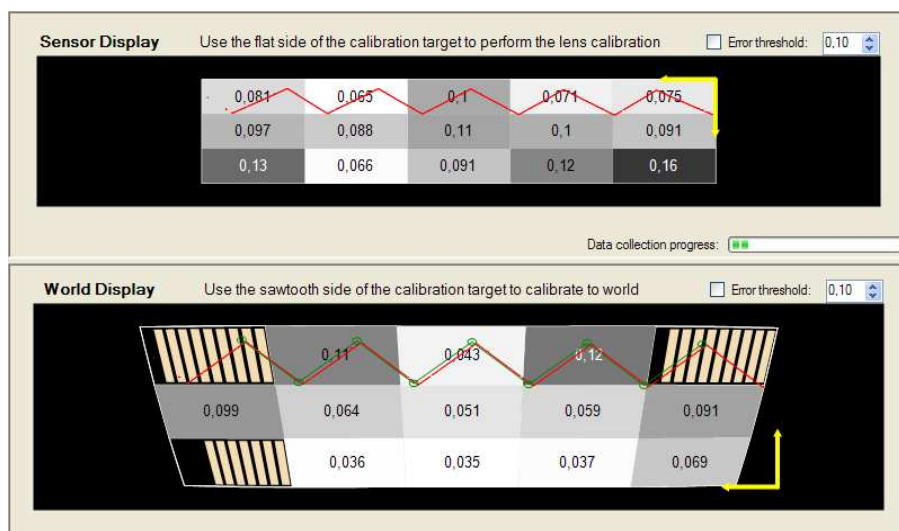


Figura 3.8: Le due finestre da utilizzare in calibrazione: la finestra più in alto (“Sensor Display”) è usata con il lato piatto del calibro, l’altra (“World Display”) con il lato a dente di sega

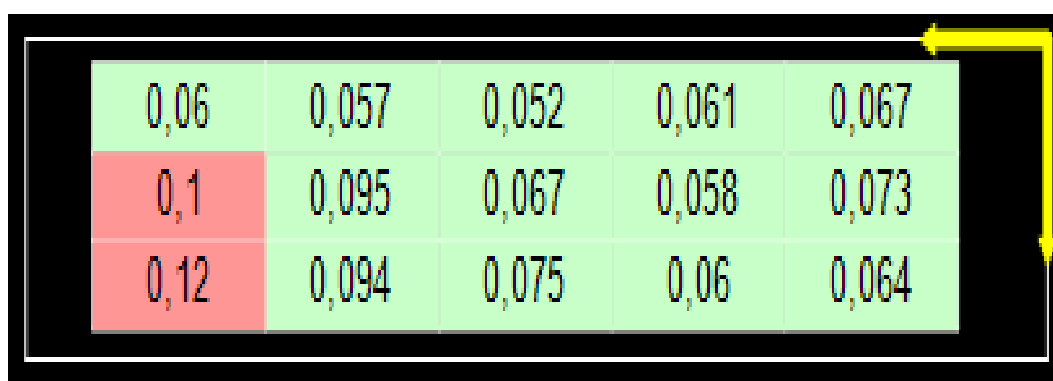


Figura 3.9: Errore di calibrazione spuntando la casella “Error Threshold”

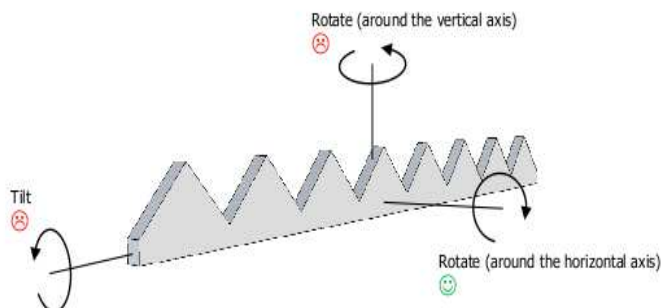


Figura 3.10: Rotazioni del calibro consentite

che il fascio laser ricada sempre sopra lo strumento di calibrazione (come mostrato in figura 3.11). Una volta cambiata la posizione del calibro si rimane fermi in modo tale che la telecamera possa acquisire un nuovo profilo. Il laser è visualizzato in rosso all'interno della finestra. Le celle della griglia intanto ad ogni acquisizione si aggiornano e si riempiono con dei rettangolini gialli, fino a quando non compare un valore, che come detto prima, rappresenta l'errore di distorsione della lente che si ha in quella cella.

Quando tutte le celle della griglia hanno il loro valore di errore, e quando questo è di un valore per noi accettabile si può passare alla seconda fase, altrimenti si può scegliere di iterare ancora finché l'errore non diventa di nostro gradimento. Nel caso che si sia soddisfatti dei valori delle celle, si può salvare il risultato della calibrazione attraverso il tasto "Save" e automaticamente si accederà alla seconda fase. Quindi nella prima fase si è compensata la distorsione introdotta dalla lente.

Nella seconda fase si ha la correzione dell'errore prospettico e bisogna utilizzare la parte a denti di sega del calibro. I risultati di questa fase sono presentati nella finestra in basso del software di calibrazione. Le azioni da compiere in questa fase sono le stesse della precedente con l'unica differenza del lato del calibro utilizzato. In questa fase, come nella precedente, la linea del laser appare sullo schermo di colore

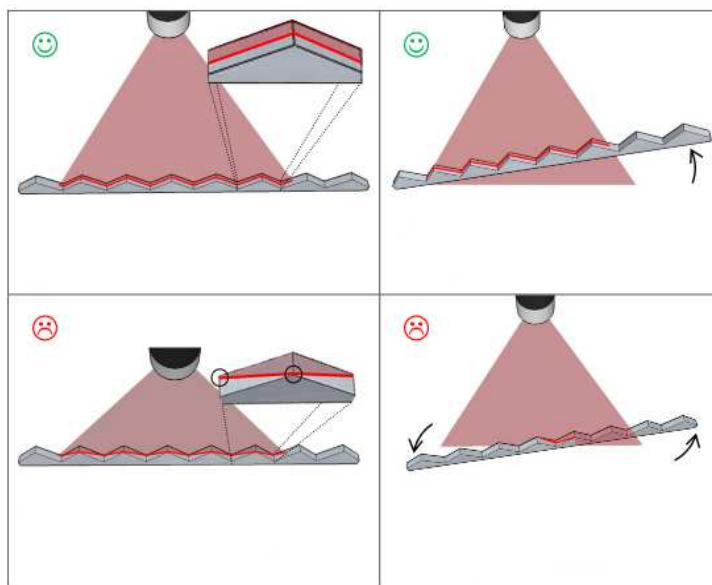


Figura 3.11: Modi corretti e non corretti di tenere il calibro

rosso, ma quando la telecamera riesce ad acquisire un profilo, essa disegna sulla ROI, in verde, la linea del laser ed i punti caratteristici cerchiati su cui ha acquisito come evidenziato in figura 3.12. I punti caratteristici sono usati per misurare la distanza fra i punti nel sistema di coordinate mondo. Da notare, nelle finestre in figura 3.8, in giallo vi è indicato il sistema di coordinate. Nella finestra “Sensor Display” è indicato il sistema di riferimento (x,z) del sensore in pixels; mentre nella finestra “World Display” è indicato il sistema di riferimento del mondo reale (x,z) in millimetri o inches (dipende dalla scelta fatta dall’utente).

Riempita anche la griglia del “World Display” si può andare avanti. Una volta fatti questi passi, si è completata la calibrazione, e quindi il software ha creato una *Look-Up Table* che può essere salvata direttamente sulla camera e/o salvata su di un file (con estensione .lut). Si consiglia di fare entrambi i salvataggi, poichè il file può essere anche caricato nella telecamera via software, cosa che risulta molto utile.

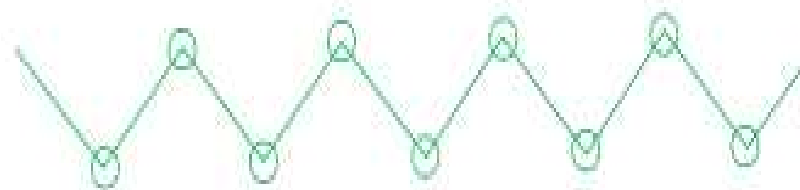


Figura 3.12: Linea del laser con cerchiati i punti caratteristici

### 3.6 L' algoritmo usato in calibrazione

In questo paragrafo viene presentato l' algoritmo di calibrazione usato nell' applicativo software *Coordinator*. Vengono inoltre fornite informazioni per la costruzione di un proprio algoritmo di calibrazione (per esempio nel caso si utilizzi una geometria diversa dalla "Reversed Ordinary").

Il software *Coordinator* determina il cambio di coordinate globali dalle coordinate del sensore alle coordinate mondo nel piano del laser e lo fa attraverso due piani 2D, il piano del sensore ed il piano dell'ottica. Non vi è dunque una mappatura 3D diretta che espliciti la traslazione e rotazione che permette di effettuare un cambiamento di coordinate tra il piano del sensore ed il piano del laser. Quindi a differenza di altri modelli geometrici qui non c'è bisogno di esplicitare tutte le caratteristiche 3D del modello, ma non si perde neanche di generalità poichè la mappatura da piano a piano riesce ad implementare esplicitamente tutti gli effetti di traslazione, rotazione e di prospettiva che si hanno nel modello 3D.

Si parte dal sistema di coordinate del sensore  $(u, v)$  dove viene usata la  $u$  per indicare le colonne e la  $v$  per indicare le righe; le unità di misura sono in pixels. All'uscita della telecamera si hanno dei dati non calibrati, si ha un  $v$  - *value* per ogni valore intero di  $u$ , ottenuto con la risoluzione dei sub-pixels (1/16 di pixels usando la componente

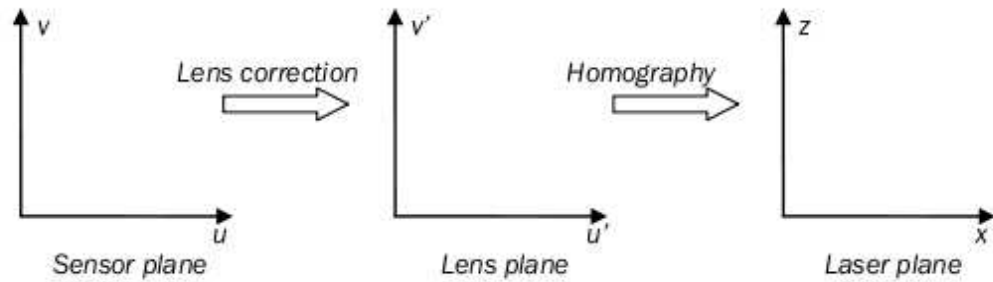


Figura 3.13: Mappatura dal piano del sensore a quello del laser

Hi3D). Ogni componente  $(u, v)$  è mappata nelle coordinate del piano del laser in due passi, esattamente i due passi che si svolgono attraverso l'interfaccia di calibrazione del software "Coordinator". Quindi si ha un primo passo in cui si ha la correzione della distorsione della lente ed un secondo in cui si esegue la correzione prospettica (Omografia).

### 3.6.1 Correzione della distorsione della lente

Per correggere la distorsione causata dalla lente il software "Coordinator" usa un modello polinomiale standard con componente tangenziale. Le coordinate  $(u', v')$  sono calcolate dalle coordinate  $(u, v)$  come segue

$$u' = u + u_0(c_1 r^2 + c_2 r^4) + 2c_3 u_0 v_0 + c_4(r^2 + 2u_0^2)$$

$$v' = v + v_0(c_1 r^2 + c_2 r^4) + 2c_4 u_0 v_0 + c_3(r^2 + 2v_0^2)$$

$$u_0 = u - u_c$$

$$v_0 = v - v_c$$

$$r = \sqrt{u_0^2 + v_0^2}$$

dove  $(u_c, v_c)$  è il centro ottico sul piano del sensore e  $c_1, c_2, c_3, c_4$  sono i parametri di distorsione della lente.

Usualmente nella descrizione di un processo di calibrazione, si dividono i parametri in gioco in due categorie: i parametri intrinseci ed i parametri estrinseci. Il centro ottico  $(u_c, v_c)$  è parte dei parametri intrinseci della telecamera. Ci sono anche altri parametri intrinseci alla camera che però si possono trascurare; quindi il centro ottico rimane l'unico parametro intrinseco da considerare.

La stima della distorsione della lente viene fatta utilizzando la parte piatta dello strumento di calibrazione; essa non ci può dare nessuna informazione sul calcolo della distorsione sullo strumento. Diventa così importante presentare lo strumento con differenti angolazioni, in modo tale da poter ricevere maggiori informazioni. Inoltre è importante fare ricevere dei profili alla telecamera con il calibro presente nelle zone periferiche della FOV, poichè è qui che la distorsione della lente è maggiore.

### 3.6.2 Omografia

Dopo aver corretto la distorsione della lente, il piano  $(u', v')$  ottenuto viene mappato nel piano del laser  $(x, z)$  usando l'omografia, cioè si determina una trasformazione in grado di mappare punti di un piano su punti di un altro piano. Questo ci garantisce che ogni punto di uno spazio corrisponde ad uno ed un solo punto dell'altro spazio.

Quindi, per definizione di omografia, si deve trovare una matrice  $H \in M^{3 \times 3}$  tale che:

$$\begin{pmatrix} X \\ Z \\ s \end{pmatrix} = H \begin{pmatrix} u' \\ v' \\ 1 \end{pmatrix} = \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{pmatrix} \begin{pmatrix} u' \\ v' \\ 1 \end{pmatrix}$$

Normalizzando con il fattore di scala  $s$ , abbiamo le coordinate mondo  $(x, z)$ :

$$x = \frac{X}{s} \quad z = \frac{Z}{s}$$

In teoria tutti i nove coefficienti dell'omografia si possono ricavare con una singola scansione del calibro dal lato a dente di sega. Però si preferisce calcolare i coefficienti usando più scansioni; in questo modo si riesce ad ottenere un modello più robusto.

### 3.6.3 Approcci alternativi

Il metodo sopra descritto non è l'unico modo per calibrare la telecamera, ma viene considerata (dalla casa costruttrice) la strada più semplice per la maggior parte delle applicazioni in cui la telecamera può essere utilizzata. È considerata la più semplice perché è veloce, non è necessario conoscere la posizione del calibro e non necessita di luci ulteriori oltre a quella del fascio laser. Inoltre questo approccio utilizza lo stesso sistema geometrico telecamera - laser di quello utilizzato nel sistema di acquisizione e dà risultati immediati sulla qualità della calibrazione. Di seguito vengono presentati alcuni approcci alternativi alla calibrazione:

- **Controllo della posizione:**

Una delle principali caratteristiche del metodo descritto sopra è che non è necessario nessun controllo sulla posizione del calibro. Essa infatti, spesso è onerosa da ottenere. Se invece si utilizza la telecamera su di un manipolatore si ha già a disposizione l'informazione sulla posizione.

Il calibro, può quindi essere collocato in varie posizioni conosciute in modo tale che la telecamera riesca ad acquisire dei profili e riesca a ricavare la correzione della lente e della prospettiva. Se l'informazione della posizione è accurata questo metodo risulta molto più potente del metodo utilizzato dal Coordinator sia perché servono poche scansioni, sia perché i parametri ottenuti risultano più robusti.

- **Calibro multi-faccia:**

La forma del calibro a dente di sega è stata scelta perché essa dà informazioni

accurate sui punti caratteristici presenti sia sull'asse x che sull'asse z. Tuttavia l'oggetto di calibrazione può essere ottenuto anche usando altre forme, oppure una combinazione di diverse forme sullo stesso oggetto di calibrazione.

Un approccio è quello di utilizzare un oggetto di calibrazione piatto con diversi cambi di pendenza differenti fra di loro. Traslando il calibro attraverso il piano del laser sono disponibili abbastanza informazioni per ottenere la mappa di calibrazione. Questo approccio comporta tuttavia diversi svantaggi rispetto al calibro a dente di sega:

- è necessario un movimento di traslazione “liscio”, senza scatti;
- il vettore di direzione della traslazione deve essere noto e tipicamente deve essere perfettamente ortogonale al piano del laser;
- è difficilmente riempibile l'intera FOV con una singola scansione e quindi sono necessarie più scansioni.

● **Stampa di pattern planari:**

Un altro metodo utilizzabile per la calibrazione della telecamera è il disegno di forme note (es. cerchio, quadrato, etc.) su di una posizione nota all'interno del piano del laser con l'ausilio di una luce bianca. In questo modo impostando la modalità della telecamera in “image” ed acquisendo una immagine 2D in scala di grigi è possibile ottenere (anche con una sola immagine) l'informazione su tutti i parametri estrinseci ed intrinseci della telecamera senza eseguire una vera e propria mappatura 3D. Questo approccio però ha diversi svantaggi rispetto all'approccio con il calibro a dente di sega:

- potrebbe essere molto complicato posizionare il pattern 2D sul piano del laser;

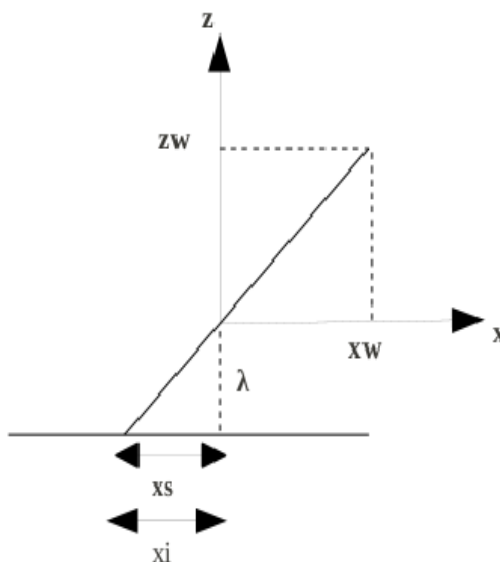


Figura 3.14: Modello Pinhole

- non è usato il fascio laser durante la calibrazione e questo non consente di correggere alcuni dei problemi che il laser comporta in fase di acquisizione;
- il filtro passa-banda della telecamera blocca la maggior parte della luce bianca, quindi visto che è usata per disegnare il pattern, bisogna allargare il tempo di apertura focale, con possibile sfocamento dell'immagine.

### 3.6.4 Coordinate mondo nell'uso senza encoder

Un altro aspetto da considerare quando si va ad utilizzare la telecamera è il calcolo delle coordinate mondo. Poichè la telecamera, non dotata di un encoder, restituisce via software solo le coordinate calibrate (quindi in millimetri)  $(x, z)$ , bisogna ricavare la  $y$  attraverso uno studio geometrico sul modello pinhole (se si possiede un encoder la  $y$  viene restituita da quest'ultimo). Il modello pinhole è rappresentato in figura 3.14. Con  $(x_w, z_w)$  si indicano le coordinate mondo, con  $x_s$  si indica la  $x$  del sensore e con  $x_i$  la  $x$  dell'immagine (praticamente l'indice di colonna). Il discorso verrà affrontato

riguardo alla coordinata  $x_w$ , ma analogamente è possibile affrontarlo per trovare la  $y_w$  che è quella che interessa ottenere. Dalla figura 3.14 si può scrivere:

$$\frac{x_w}{z_w} = \frac{x_s}{\lambda}$$

dove  $\lambda$  è la lunghezza focale. Inoltre vale anche la seguente relazione:

$$\frac{x_i}{W_i} = \frac{x_s}{W_s}$$

dove  $W_i$  è la risoluzione massima dell'immagine lungo l'asse  $x$ , che nel caso della telecamera di range Sick Ranger D50 risulta essere di 1536 pixels, mentre  $W_s$  è la dimensione del sensore lungo l'asse  $x$  (larghezza) che risulta essere 14,6 mm. Dalle due relazioni si evince che

$$x_w = \frac{x_s}{\lambda} \cdot z_w = \frac{z_w}{\lambda} \cdot \left( \frac{x_i}{W_i} \cdot W_s \right) = \frac{x_i \cdot z_w}{\lambda'_x}$$

dove

$$\lambda'_x = \frac{W_i}{W_s} \cdot \lambda$$

è una lunghezza focale normalizzata. Lo stesso discorso si applica alla coordinata  $y$ , sostituendo la coordinata con la  $y$  e la larghezza del sensore ed immagine con la lunghezza degli stessi. Per la  $y_w$  si trova la seguente relazione:

$$y_w = \frac{y_s}{\lambda} \cdot z_w = \frac{z_w}{\lambda} \cdot \left( \frac{y_i}{H_i} \cdot H_s \right) = \frac{y_i \cdot z_w}{\lambda'_y}$$

dove indichiamo con  $y_s$  la  $y$  del sensore, con  $y_i$  la coordinata  $y$  dell'immagine (indice di riga) e con  $H_i$ ,  $H_s$  rispettivamente la risoluzione massima dell'immagine lungo la  $y$ , che sappiamo essere di 512 pixels, e la dimensione del sensore lungo l'asse  $y$ , che sappiamo essere di 4,9 mm. Anche la lunghezza focale normalizzata si trova come nel

precedente caso:

$$\lambda'_y = \frac{Hi}{Hs} \cdot \lambda$$

Disponendo di tutti i dati, si è in grado ora di calcolare le due lunghezze focali normalizzate:

$$\lambda'_x = \frac{1536}{14,6} \cdot 25 = 2630,1369$$

$$\lambda'_y = \frac{512}{4,9} \cdot 25 = 2612,2448$$

considerando come lunghezza focale quella dell'ottica montata sulla telecamera presente in laboratorio (25 mm).

## Capitolo 4

# Apparato sperimentale e sistema di acquisizione

### 4.1 Introduzione

In questo capitolo viene illustrato l'apparato sperimentale costruito per l'uso della telecamera insieme ad una sua analisi in cui verranno usate le relazioni presentate nel capitolo 2. Inoltre verranno descritte le varie modalità di acquisizione, i parametri della telecamera più importanti che è necessario modificare a seconda dell'applicazione in cui viene utilizzata e l'uso delle API messe a disposizione dalla casa costruttrice per l'acquisizione dei dati tridimensionali.

Infine, verranno introdotti dei risultati di misure di ripetibilità eseguite con la telecamera su oggetti diversi e un confronto con il laser *2D CMOS Serie ZS-L* movimentato dal robot *AlphaCU4* introdotto nel capitolo 1 e illustrato in [7] sulle misurazioni eseguite su una moneta di un euro.

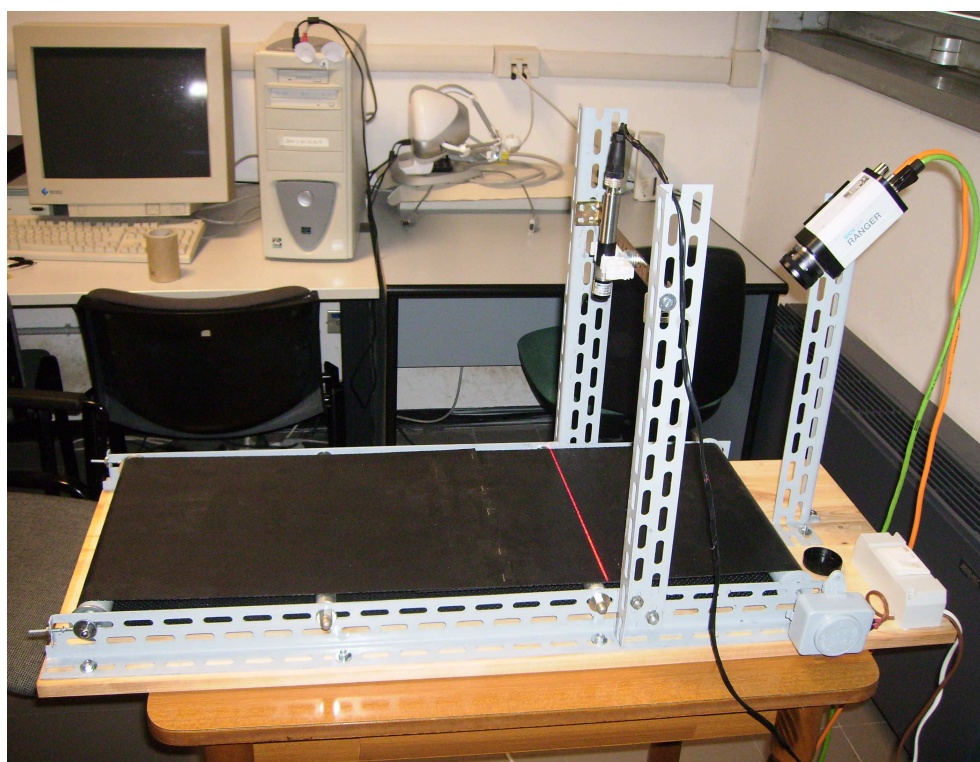


Figura 4.1: Apparato sperimentale costruito per il laboratorio

## 4.2 Apparato sperimentale e sua analisi

L'apparato sperimentale (mostrato in fig. 4.1) costruito per il laboratorio è, in scala ridotta, il setup standard per il quale la telecamera è stata concepita, ossia un nastro trasportatore sopra il quale sono collocati telecamera ed illuminatore laser. In questo caso, avendo scelto di usare la geometria "Reversed Ordinary", il laser è posto sulla normale al piano di lavoro e la telecamera forma un angolo  $\alpha$  con esso. Essendo un setup di tipo sperimentale e dovendolo usare in laboratorio per poter testare la telecamera in varie situazioni, la principale caratteristica dell'apparato costruito è la flessibilità. Infatti, come si può notare dalla figura 4.1, il sistema è stato realizzato con delle piccole lastre di alluminio traforate ad intervalli regolari. Questo permette di avere uno strumento modulabile nella configurazione in base alle proprie esigenze.

Infatti la struttura che sostiene il laser può essere spostata liberamente lungo l'asse di movimento ed il laser può essere spostato lungo la normale più vicino o più lontano dal piano di lavoro. La telecamera, invece, può essere spostata solo lungo la normale (o a destra e sinistra se si fanno degli altri fori nel legno).

I supporti per il nastro sono stati costruiti con dei tubi, due esterni che effettuano la rotazione e due centrali di supporto, saldando all'estremità delle rondelle per contenere il nastro all'interno degli stessi. La rotazione è data da un motorino di un grill, l'unico motorino a disposizione, purtroppo. Se vi è la possibilità è meglio utilizzare un motorino più veloce, anche perchè la telecamera lo permette.

Il nastro è realizzato con della gomma antiscivolo nera che si trova in un comune negozio di bricolage. Se vi è la possibilità sarebbe meglio trovare un vero e proprio nastro utilizzato negli strumenti professionali, poichè si è notato che la gomma è un materiale che può dare dei problemi alla telecamera a causa dell'assorbimento del fascio laser. Il problema principale, derivante dal nastro, è il rumore presente nelle immagini di range, che comunque è stato facilmente eliminato con un filtro.

### 4.2.1 Analisi dell'apparato sperimentale

Ora si vuole effettuare un'analisi simile a quella effettuata su un esempio nel capitolo 2. I dati necessari, ricavabili dal setup a nostra disposizione (mostrato in fig. 4.1) e dal datasheet di ottica e telecamera sono i seguenti:

- **Baseline (B):** 230 mm
- **Altezza laser (R):** 280 mm
- **Working distance (WD):** 360 mm
- **Risoluzione sensore:**  $1536 \times 512$  pixels
- **Dimensione sensore:**  $14,6 \times 4,9$  mm

- **Lunghezza focale ( $\lambda$ ): 25 mm**

Usando l'equazione 2.1 si ha che la larghezza del FOV ottenuta è:

$$FOV_{width} = \left( \frac{Sensor_{width}}{FocalLength} \right) \cdot WorkingDistance = \left( \frac{14,6}{25} \right) \cdot 360 = 210,24 [mm]$$

La stessa equazione si può riscrivere in base all'altezza del sensore, per trovare l'altezza del FOV:

$$FOV_{height} = \left( \frac{Sensor_{height}}{FocalLength} \right) \cdot WorkingDistance = \left( \frac{4,9}{25} \right) \cdot 360 = 70,56 [mm]$$

Ora si vuole calcolare l'angolo presente tra la telecamera ed il laser. Si può calcolare utilizzando l'equazione 2.11:

$$\alpha = \arcsin\left(\frac{B}{WD}\right) = \arcsin(0,6389) = 39,71^\circ$$

Si possono inoltre ricavare le varie risoluzioni lungo gli assi, usando l'equazione 2.2 per quanto riguarda l'asse  $x$ , mentre per l'asse  $z$  vengono usate le equazioni presentate nella tabella 2.2:

$$X_{res} = \frac{FOV_{width}}{numeroDiPixel} = \frac{210,24}{1536} = 0,1368 [mm/pixel]$$

$$Z_{res} = \Delta Z = \frac{X_{res}}{\sin(\alpha)} = \frac{0,1368}{\sin(39,71^\circ)} = 0,2294 [mm/pixel]$$

Per calcolare la risoluzione lungo l'asse  $y$  bisognerebbe sapere la velocità di movimento del nastro, dato che però non è in nostro possesso. Nel caso si conoscesse e si lavorasse in modalità "free" si potrebbe utilizzare l'equazione 2.3, mentre nel caso si utilizzasse un encoder si potrebbe utilizzare l'equazione 2.4.

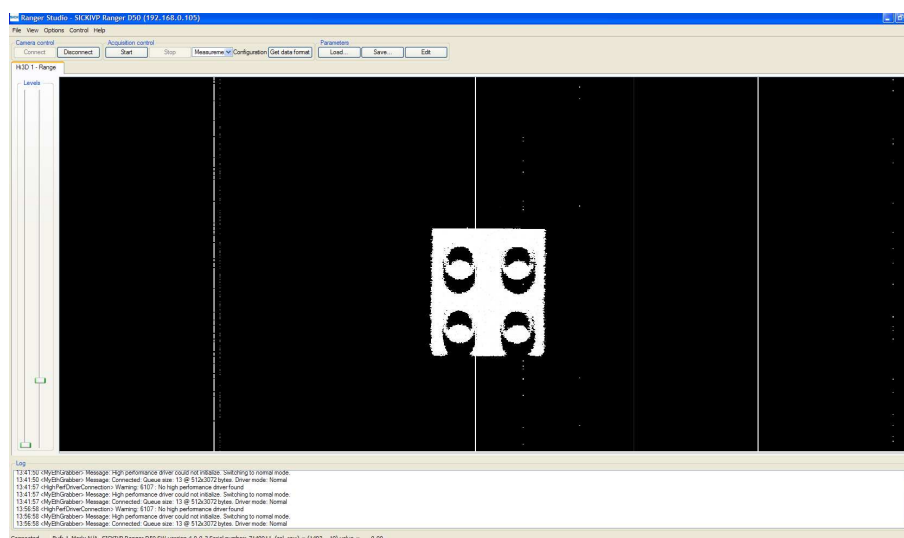


Figura 4.2: Un mattoncino scansionato con Ranger Studio

### 4.3 Il programma “Ranger Studio”

Con la telecamera, la Sick dà in dotazione anche un programma di acquisizione di immagini e misurazioni, il “Ranger Studio”. Su questo software è possibile scegliere la modalità di utilizzo “image” o “measurement”, in modo tale che si possono acquisire immagini o misure di un oggetto. Ovviamente la modalità più interessante, almeno dal nostro punto di vista, è la modalità “measurement”.

Da questo programma si possono anche modificare i parametri della telecamera e salvarli in un file, oppure caricare dei parametri già salvati. Oltre alla visualizzazione in scala di grigi, è disponibile un visualizzatore 3D che però dà risultati pessimi. È stao quindi necessario creare un visualizzatore in OpenGL per sopperire a questa mancanza. Vi è la possibilità tramite le barrette esterne alla finestra di visualizzazione di passare gradualmente da una immagine a scala di grigi ad una in bianco e nero, oppure di aumentare o diminuire il contrasto all’interno dell’immagine.

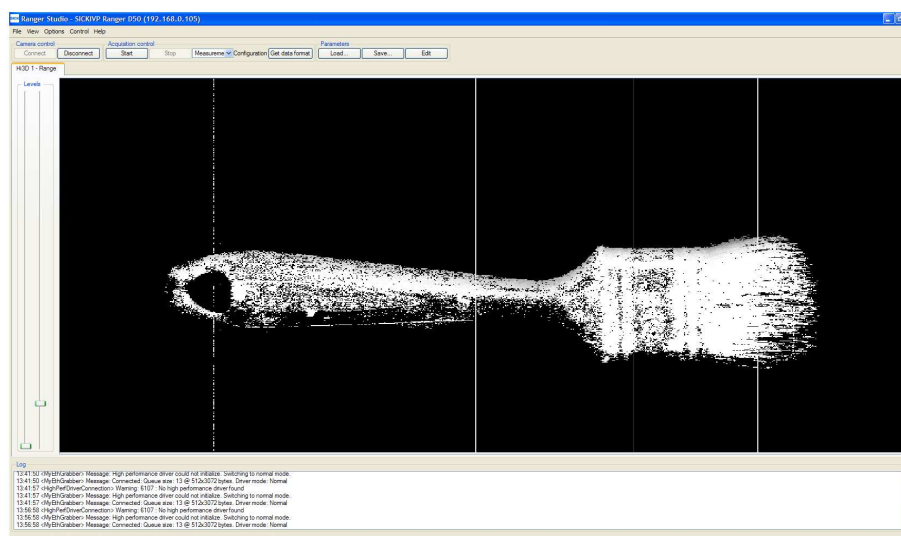


Figura 4.3: Un pennello scansionato con Ranger Studio

## 4.4 Parametri della telecamera: modalità “measurement”

La configurazione “measurement” è usata per fare le misurazioni in 3D di oggetti ed ha una sola componente di misura, l’“High-Resolution 3D” (DCM) e nella modalità “free” è limitato all’acquisizione di 1000 profili per secondo. Di seguito faremo un elenco con relativa spiegazione dei parametri più importanti ma non gli unici presenti nella telecamera. Per tutti gli altri si rimanda al manuale della SICK.

- **Encoder direction:** Specifica quale direzione è intesa come “in avanti”. Il parametro è molto importante se il parametro **Trig mode** è impostato a 3 o 4. Le due opzioni del parametro sono le seguenti:
  - **0:** Standard, cioè la direzione “in avanti” è intesa la direzione dell’asse y positivo, rispetto al sistema di riferimento (vedi paragrafo 2.3).
  - **1:** Reversed, cioè la direzione “in avanti” è quella delle y negative.

- **Trig mode:** Questo parametro indica in che modalità di triggering si vuole scansionare. Le opzioni per questo parametro sono le seguenti:
  - **0:** Free running mode/No triggering. Se il parametro è settato a “0” allora vuol dire che ci si trova in modalità “free”, cioè senza encoder.
  - **2:** Pulse Triggering. Se settato a “2” allora la telecamera effettua la scansione dell’oggetto solo se l’oggetto si trova in una posizione in cui non è stato scansionato in precedenza. Questa modalità risente molto delle vibrazioni ma può essere usata una singola fase dell’encoder.
  - **3:** Position mode. Come l’opzione “2” ma in questa modalità vi è robustezza rispetto alle vibrazioni.
  - **4:** Direction mode. Scansiona quando l’oggetto si muove in avanti. La direzione è data dal parametro **Encoder direction**. Modalità robusta alle vibrazioni.
  - **5:** Motion mode. Scansiona l’oggetto ogni qualvolta si muove, non importa in quale direzione. Robusta alle vibrazioni.
  - **Valore di default:** 0.
- **Cycle time:** Parametro estremamente importante che indica il tempo fra l’acquisizione di un profilo ed un altro. Il suo valore va impostato in base al fatto che si utilizzi un encoder o no, quindi in base al valore del parametro **Trig mode**:
  - Se **Trig mode = 0:** in questo caso indica il tempo in microsecondi fra ogni misurazione;
  - Se **Trig mode = 2:** in questo caso indica il più piccolo tempo possibile fra 2 misurazioni.

Il valore di default è di 3000 microsecondi. Il valore minimo e massimo impostabili sono rispettivamente 1000 e 50000 microsecondi. In realtà il valore

minimo impostabile nella telecamera in nostro possesso in laboratorio è 5000 microsecondi, non si riesce ad impostare un valore più basso di questo. Il valore impostato per le applicazioni sviluppate in questa tesi è stato il valore massimo, ossia 50000 microsecondi.

- **Start column:** Indica la prima colonna del sensore sulla quale acquisire i dati. Sulla telecamera disponibile, la Ranger D50, questo valore deve cadere nel range 0-1280 (si ricorda che il numero totale di colonne è di 1536).
- **Number of columns:** Indica il numero di colonne sulle quali acquisire i dati. Questo valore **deve** essere un multiplo di 8. Per la telecamera disponibile questo valore deve cadere nel range 256-1536, il valore di default è 1536.

Questi due parametri (**Start column** e **Number of columns**) sono utili quando non si vuole acquisire l'intera immagine ma solo una parte di essa o una parte particolare dell'oggetto.

- **Mark:** Specifica se includere o no le informazioni provenienti dall'encoder nelle scansioni.
  - **0:** non include i valori provenienti dall'encoder.
  - **1:** include i valori nella scansione.
- **Mark with:** Specifica quali valori inviare al mark; il numero di impulsi provenienti dall'encoder oppure il numero di scansioni eseguite dalla telecamera. I possibili valori sono:
  - **0:** fa il mark con gli impulsi del segnale di encoder.
  - **1:** fa il mark con il numero di scansioni fatte dalla telecamera.
  - **2:** fa il mark riflettendo la posizione dell'encoder. Incrementa il mark se l'encoder si muove nella direzione impostata come "in avanti" dal parametro

**Encoder direction**, mentre decrementa se si muove nella direzione opposta. Permette un tracking preciso su oggetti che si muovono in entrambe le direzioni.

- **3**: fa il mark riflettendo il movimento dell'encoder. Esso incrementa indipendentemente dalla direzione di movimento, ma permette un tracking preciso della distanza percorsa da un oggetto.
- **Valore di default**: 0.

#### 4.4.1 High-resolution 3D (DCM)

Il componente High-resolution 3D misura il range utilizzando un algoritmo simile al centro di gravità e misura il range con una risoluzione di 1/16-esimo di pixel. I seguenti parametri possono essere settati per configurare questa componente:

- **Enable**: Specifica se attivare o disattivare la componente nelle scansioni.
  - **0**: disattivata;
  - **1**: attivata.
- **Start row**: Specifica la prima riga del sensore dalla quale iniziare ad acquisire. I valori possono andare da 0 a 511. Il valore di default è 0.
- **Number of rows**: Specifica il numero di righe per le quali fare l'acquisizione. I valori possibili sono da 16 a 512; il valore di questo parametro **deve** essere un multiplo di 16. Valore di default: 512.
- **Exposure time**: indica il tempo in microsecondi durante il quale il sensore è esposto alla luce. Il range è da 10 a 50000. Valore di default: 5000.

Tutti i sopracitati parametri possono essere modificati, salvati e caricati dal software "Ranger Studio" (vengono salvati su file con estensione ".prm") oppure attraverso

l'utilizzo delle API. Nel paragrafo 4.6 è spiegato come cambiare i precedenti parametri utilizzando una classe opportunamente realizzata per acquisire dalla telecamera.

## 4.5 Icon API

Installando il software in dotazione con la telecamera Sick Ranger D50, si installano, in realtà, due differenti tipi di API:

- **Icon C++ API:** sono le API create per essere usate in linguaggio C++;
- **Icon C API:** sono le API per il linguaggio C.

Per realizzare le applicazioni sviluppate in questa tesi, così come la classe `Acquisizione` illustrata nel paragrafo 4.6, sono state usate le API per l'ambiente C++. Nella spiegazione e descrizione delle API, si farà riferimento alle API in C++, ma un discorso analogo vale per le API in C, poichè le API hanno le stesse funzioni e cambia solo la sintassi del linguaggio.

Le Icon API sono basate su due classi principali:

- **Camera:** classe usata per controllare, avviare o interrompere l'acquisizione della telecamera o cambiare i suoi parametri;
- **Grabber:** classe su cui l'applicazione deve contare per ricevere i dati delle misurazioni. Nelle applicazioni si possono utilizzare delle sottoclassi che corrispondono al tipo di telecamera che si usa:
- **EthernetCamera:** per la Ranger D/E e la Ruler E;
- **RangerC:** per la telecamera RangerC.

Le API funzionano solo in ambiente *Windows Xp*. Quando si costruisce un'applicazione occorre ricordarsi di linkare le librerie Icon.

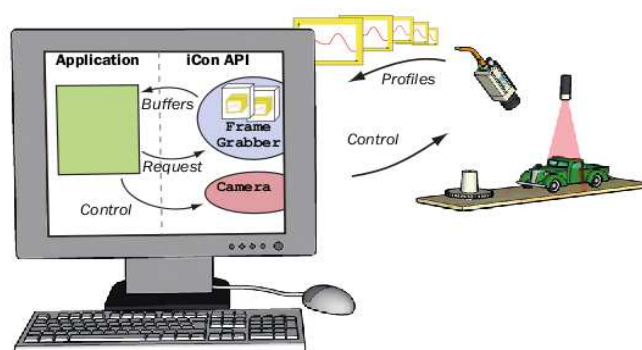


Figura 4.4: Utilizzo delle classi *Camera* e *Grabber* nelle applicazioni

## 4.6 Classe di acquisizione

Per agevolare l'acquisizione è stata realizzata una classe di acquisizione di uso generale, dove è possibile avviare o interrompere la telecamera ed anche cambiare o caricare i parametri da un file sulla telecamera. Inoltre si può avviare il *frame grabber* e ricevere i dati dalla telecamera già calibrati (supponendo di avere già il file *.lut* di calibrazione) sotto forma di array (della dimensione della risoluzione scelta,  $HEIGHT \times WIDTH$ ) oppure sotto forma di insiemi di punti, utilizzando una libreria che permette di gestire questo tipo di dati, come la *PCL* (Point Cloud Library).

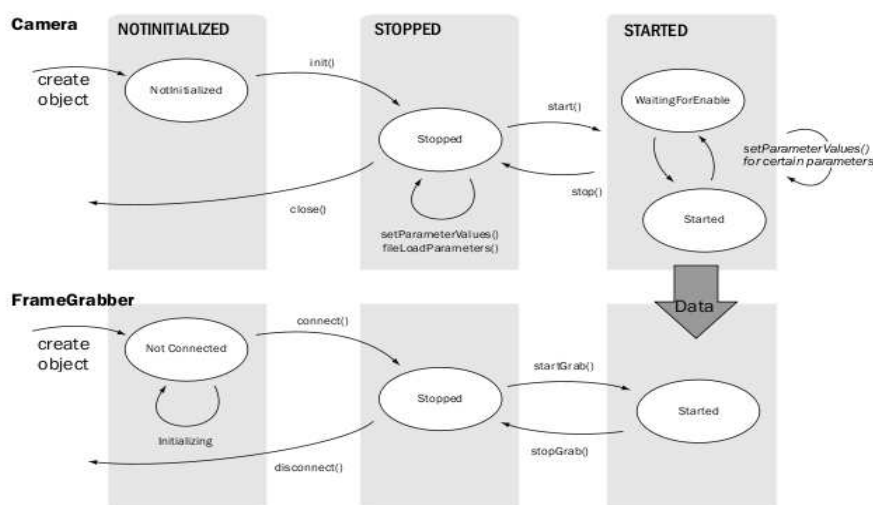
Gli stati in cui si possono trovare *camera* e *frame grabber* sono illustrati in fig 4.5. Vediamo ora le principali funzioni della classe *Acquisizione*:

- **Acquisizione(string IP):**

è uno dei costruttori della classe. In questo viene passato solo l'indirizzo IP a cui è connessa la telecamera, tutti i parametri sono quelli di default. Se si vuole cambiare i parametri singolarmente lo si può fare attraverso la funzione *setParameterValue()* delle Icon API;

- **Acquisizione(string IP, string PathFileParametri):**

è un altro costruttore della classe, il più comodo poichè consente di connetter-


 Figura 4.5: Diagramma degli stati di *camera* e *frame grabber*

si alla camera attraverso l'indirizzo IP passato e permette di caricare un file dei parametri già esistente passando il suo path come parametro (*PathFileParametri*); I due costruttori creano i due oggetti *camera* e *frame grabber*. Settano l'IP della telecamera e se specificato settano anche i parametri; inoltre caricano il file di calibrazione (se presente nella telecamera) e impostano la modalità su "measurement". Dopodichè avviano il frame grabber.

- **StartCamera():**

dopo aver creato l'oggetto *Acquisizione*, chiamando questa funzione si avvia la telecamera.

- **StopCamera():**

stoppa la telecamera.

- **SetTimeout(unsigned int time):**

questa funzione è il setter del timeout; cioè del tempo che la telecamera attende i dati prima di andare in timeout. Il tempo che si passa alla funzione si intende in

millisecondi (per esempio se  $time = 3000$ , il tempo di timeout sarà di 3 secondi).

Il relativo getter è la funzione *GetTimeOut()*.

- **SetCameraMode(string mode):**

questa funzione serve a settare la modalità di funzionamento della camera, se *mode* è “Measurement” ci si trova nella modalità di misurazione (opzione di default) , mentre se *mode* è “Image” ci si trova nella modalità di acquisizione delle immagini. Ovviamente vi è il relativo getter *GetCameraMode()*.

- **SetCameraIp(string ipAdress):**

è il setter dell’indirizzo IP della telecamera. Prima di settare l’indirizzo IP, nel caso che la telecamera sia già avviata, ricordarsi di stopparla; dopo riavviarla usando le relative funzioni. Il getter è implementato con la funzione *GetCameraIp()*.

- **SetParameterFile(string parameterFilePath):**

questa funzione setta il file dei parametri da caricare. Vale lo stesso discorso dell’indirizzo IP, prima di chiamare la funzione, va stoppata la telecamera e dopo riavviata. Il parametro da passare è il path del file dei parametri che si vuole caricare. Il relativo getter è la funzione *GetParameterFile()*.

Le funzioni principali che permettono di acquisire l’immagine di range sono illustrate di seguito:

- **GetRangeImageInVector(vector<float> rangeVector):**

alla funzione va passato un *vector* di *float* che verrà riempito con i valori di range misurati.

Si può accedere all’elemento nel posto  $(i, j)$  accedendo al vettore in questo modo:

$$element_{ij} = rangeVector[j + i * WIDTH];$$

dove  $j$  rappresenta l'indice di colonna,  $i$  l'indice di riga e  $WIDTH$  è la dimensione di larghezza utilizzata per la risoluzione.

- **GetRangeImageInPointCloud(PointCloud<PointXYZ> cloud):**

alla funzione va passato una PointCloud (PCL, Point Cloud Library) di tipo PointXYZ, che praticamente è formato da tre float, per indicare un punto nello spazio. Passando la point cloud dopo l' $i$ -esimo elemento si può leggere in questo modo:

$$element_x[i] = cloud.points[i].x;$$

$$element_y[i] = cloud.points[i].y;$$

$$element_z[i] = cloud.points[i].z;$$

Questi metodi ritornano il valore di range calibrato, ma le coordinate  $(x,y)$  rappresentano l'indice di colonna e di riga del sensore. Per ottenere i valori reali in coordinate mondo di  $x$  e  $y$  si possono utilizzare le relazioni illustrate nel paragrafo 3.6.4. Questa classe è stata scritta per l'uso senza encoder; nel caso se ne utilizzi uno, si deve apportare qualche modifica al metodo *GetRangeImageInPointCloud* per ottenere la coordinata  $y$  dall'encoder.

## 4.7 Acquisizione dati

In questo paragrafo si vuole entrare più nel dettaglio del formato dati usato dalle Icon API per acquisire dalla telecamera e delle modalità di acquisizione e filtraggio dei dati.

### 4.7.1 Icon Buffers, Scans, Profiles e Data Format

La classe di acquisizione vista in precedenza ottiene i dati dalla telecamera grazie al frame grabber nella forma di *IconBuffers*. Ogni Icon Buffer contiene un numero con-

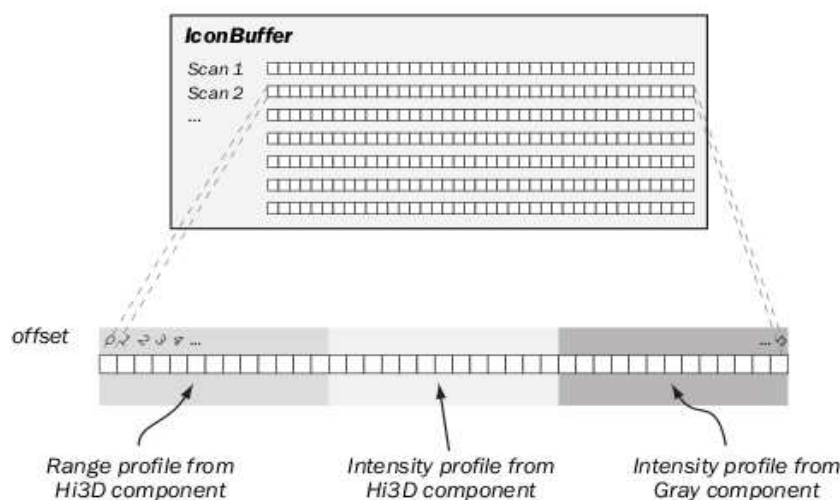


Figura 4.6: Un esempio di *IconBuffer* con la componente di range, intensità e scala di grigi

secutivo di scansioni (*scans*). Ogni scansione contiene uno o più profili (*profiles*). Il numero di profili che la scansione contiene dipende dalla configurazione della telecamera, se per esempio si usa la telecamera in nostro possesso (Ranger D50) con la componente Hi3D allora la scansione contiene un solo profilo, quello di range. Se invece la telecamera fosse la Ranger E con l'uso della stessa componente avremmo due profili, uno per il range e l'altro per l'intensità. La figura 4.6 illustra il formato dati. L'ordine con cui sono disposti i profili non si conosce a priori, e il formato dati (byte, word o float) dipende solo dalla configurazione della telecamera al momento dell'acquisizione.

L'ordine ed il formato dati delle acquisizioni è descritto dal *data format*, che può essere ritornato dalla telecamera. il *data format* può essere passato all'oggetto *FrameGrabber* che lo trasforma nell'oggetto *DataFormat* che può essere usato per localizzare un profilo all'interno di una scansione.

L'oggetto *DataFormat* contiene l'oggetto *Component* che corrisponde ai tipi di misurazioni attivati nella configurazione (Hi3D, Gray,...) ed un oggetto *SubComponent* che corrisponde al tipo di profilo (Range, Intensity,...), come mostrato in figura 4.7.

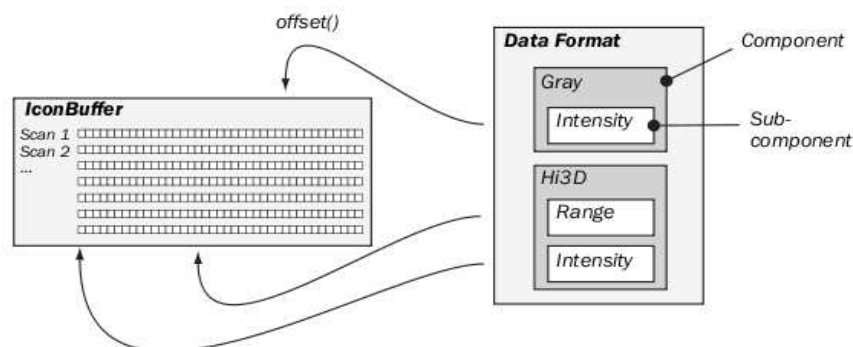


Figura 4.7: Data Format

### 4.7.2 Polling e Call-back

Vi sono due modi per ottenere l'Icon buffer dal frame grabber: attraverso la metodologia *polling* o attraverso una funzione di *call-back*.

La modalità *polling* è quella usata nella classe `Acquisizione` descritta prima. La metodologia usa la funzione `getNextIconBuffer()` per farsi restituire il buffer dal frame grabber. Prima però di chiamare questo metodo si usa la funzione `availableBuffers()` per capire se vi è un buffer pieno da acquisire oppure no. Nel caso che non si utilizzi quest'ultimo metodo e si chiamasse `getNextIconBuffer()` con nessun buffer disponibile la telecamera andrebbe in timeout dopo il tempo specificato nei parametri. Se si imposta il parametro *timeout* a 0 allora il metodo ritorna immediatamente se trova un buffer vuoto, mentre se si imposta a -1 il metodo non ritorna mai finchè non riceve un buffer pieno da acquisire. Nella classe `Acquisizione` illustrata prima, si usa il metodo `availableBuffers()` quindi questo problema non si pone.

Se invece si usa una funzione di *call-back* allora il frame grabber chiama questa funzione appena vi è un buffer disponibile. Il frame grabber poi non richiama più la funzione finchè essa non ritorna, anche se vi è un altro buffer disponibile. La funzione di *call-back* va specificata quando si crea l'oggetto `FrameGrabber`.

### 4.7.3 Filtro di calibrazione

Nella classe `Acquisizione` per ottenere i dati calibrati si utilizza il filtro di calibrazione presente nelle Icon API. Il filtro necessita della look-up table (file `.lut`) calcolata attraverso la calibrazione con il software “Coordinator” o in uno dei modi alternativi descritti nel paragrafo 3.6.3.

Nelle Icon API questo filtro è denominato *calibrationFilter* e restituisce le coordinate  $(X, R)$ , dove  $X$  è la coordinata reale dell’asse  $x$  mentre  $R$  indica il range;  $R$  e  $Z$  sono presi separatamente per indicare che il primo è parallelo al piano del laser, il secondo è perpendicolare al piano di lavoro.

Nella configurazione geometrica usata in laboratorio, la “Reversed Ordinary”,  $R$  e  $Z$  coincidono.

## 4.8 Misure

Si è voluto testare l’accuratezza delle misurazioni effettuate dal sensore. Sono pertanto state eseguite delle prove di *ripetibilità* su una serie di oggetti, mostrati in figura 4.8. Si tratta di oggetti molto differenti fra loro, infatti si ha un mattoncino Lego, un portachiavi, un fermaporta, una scatola ed un pennello. Sono state effettuate 100 scansioni diverse tenendo ciascun oggetto fermo sotto la linea del laser; in pratica si è scansionata sempre la stessa riga dell’immagine di range. Sono poi state calcolate le deviazioni standard per ogni pixel dell’immagine di range (ossia per ognuno dei  $1536 \times 512$  pixels), ed infine, per avere un dato sintetico, si è effettuata la media di tutte queste deviazioni standard. I risultati sono riassunti nella tabella 4.1. Come si può notare la deviazione standard media dei vari oggetti è molto bassa, tipicamente inferiore al millimetro. Quindi si può concludere, con i dati a nostra disposizione, che la telecamera potrebbe avere una risoluzione inferiore al millimetro. Questo risultato non può essere garantito perchè bisognerebbe effettuare delle vere e proprie scansioni sempre sulle



Figura 4.8: Oggetti usati per effettuare le misure

Oggetto	$\bar{\sigma}$ [mm]
Lego	0,573
Portachiavi	0,540
Fermaporta	0,998
Pennello	1,883
Scatola	0,302

Tabella 4.1: Deviazione standard media nei vari oggetti

stesse righe e colonne dell'immagine di range e calcolare la deviazione standard su di esse. Non è stato possibile effettuare tale prova per la mancanza di un encoder nella configurazione attuale del sistema.

Una ulteriore misura, per testare le potenzialità della telecamera e soprattutto la precisione, è stata quella di confrontare i risultati avuti con la telecamera con quelle effettuate dal sensore laser *2D CMOS Serie ZS-L*, citato nel capitolo 1 e descritto in [7], sulle misure di una moneta di 1 euro (figura 4.9). In particolare è stato calcolato lo spessore medio della moneta e il diametro della stessa. È stato considerato anche il tempo necessario per l'acquisizione dell'immagine.

Nella tabella 4.2 sono riportate le misure ufficiali della moneta, quelle effettuate dal sensore laser con indicato l'errore percentuale ed infine le misure effettuate con la



Figura 4.9: Moneta da 1 euro

telecamera con i relativi tempi di acquisizione. Sia sull'immagine acquisita dalla tele-

	Misure Ufficiali [mm]	M. Laser [mm]	M. Telecamera [mm]
<b>Spessore</b>	2,33	2,14	2,48
<b>Diametro</b>	23,25	23,00	26,05

Tabella 4.2: Misure effettuate sulla moneta da 1 euro

camera sia su quella acquisita dal sensore laser, prima di calcolare lo spessore ed il diametro, si è effettuata un'operazione di filtraggio per eliminare gli effetti di bordo ed il rumore presente nella scena. La telecamera ha un errore percentuale sullo spessore della moneta del 6,43% mentre il laser ha un errore maggiore, ossia dell'8,24%. Sul diametro invece il laser ha un errore dell'1,27% mentre quello della telecamera è del 12,04%. L'errore del laser è dovuto al fatto che possiede un passo di 1 mm, quindi non riesce a misurare lo scarto di 0,25 mm. L'errore considerevole della telecamera è dovuto probabilmente alla mancanza di un encoder. Sarebbe opportuno effettuare anche il confronto fra questa misurazione della telecamera ed un'altra effettuata con encoder, per capire se la causa dell'errore è veramente quella ipotizzata.

La differenza nei tempi di acquisizione è notevole, visto che il laser è montato su un apparato per la sua movimentazione che impiega circa 15 minuti per compiere l'acquisizione mentre la telecamera solo 25 secondi. Il tempo della telecamera è ulteriormente migliorabile, utilizzando un nastro trasportatore con maggiore velocità. Il tempo potenziale potrebbe essere dell'ordine di pochi secondi. In figura 4.10 è mostrato il cluster dei punti della moneta da 1 euro acquisita attraverso la telecamera *Sick Ranger*

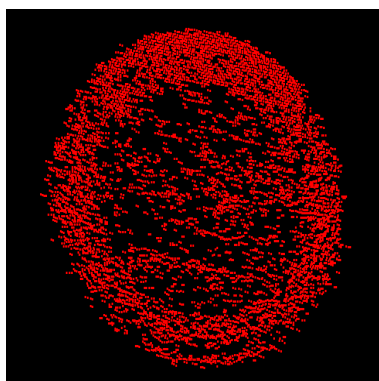


Figura 4.10: Cluster dei punti della moneta da 1 euro vista dalla telecamera *Sick Ranger D50*

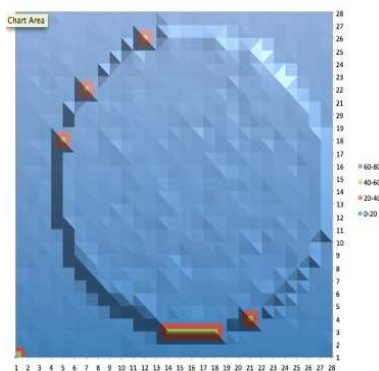


Figura 4.11: Immagine di profondità elaborata dal laser *2D CMOS Serie ZS-L*. Immagine presa da [7]

*D50* mentre in figura 4.11 è mostrata l'immagine di prossimità acquisita attraverso il sensore laser *2D CMOS Serie ZS-L*.

## Capitolo 5

# Elaborazione di immagini di profondità

### 5.1 Introduzione

Dopo aver trattato l'acquisizione di immagini di profondità ed aver misurato l'accuratezza e la precisione delle distanze misurate, ci si è posto l'obiettivo di valutare la qualità del dato acquisito dal sensore quando viene impiegato nella risoluzione di un problema. Un'applicazione interessante è il riconoscimento e la classificazione di oggetti osservati con la telecamera di prossimità. La classificazione, in generale, presuppone che sia la forma complessiva dell'oggetto sia i dettagli siano ben riconoscibili. Questa considerazione è ancora più valida se il riconoscimento viene effettuato a seguito dell'estrazione di punti caratteristici o feature. I metodi basati su estrazione di caratteristiche, resi popolari dalla diffusione di algoritmi come il SIFT (Scale invariant feature transform) [8], hanno come risultato un insieme di punti cui sono associati dei valori, solitamente denominati descrittori, che dipendono dalle caratteristiche locali e globali di un intorno dell'immagine. Sebbene la letteratura si sia occupata prevalentemente di feature per immagini a toni di grigio o a colori, ossia sprovviste di informazioni

tridimensionali, recentemente sono stati proposte delle caratteristiche sviluppate appositamente per le immagini di profondità, le caratteristiche NARF (Normal Aligned Radial Feature), descritte in [17]. Queste caratteristiche fanno parte di un progetto più ampio, la libreria PCL (Point Cloud Library), descritta in [13] e disponibile al sito <http://www.pointclouds.org>. Tale libreria permette la gestione e l'elaborazione di insiemi di punti tridimensionali.

In questo capitolo, quindi, viene descritta la creazione e l'elaborazione di immagini di prossimità, con particolare attenzione all'estrazione di feature. Inoltre vengono illustrati i vari approcci che si sono sviluppati per realizzare il classificatore di oggetti.

## 5.2 Strutture dati per immagini di range

La libreria PCL fornisce una classe che permette di contenere una immagine di prossimità, la classe è denominata `RangeImagePlanar`. Per definire quindi l'immagine di range in PCL è stata realizzata una classe `CreatePlanarImage`, che permette di creare una range image da un file PCD (Point Cloud Data) o da un oggetto *PointCloud* di tipo *PointXYZ*. Nella libreria PCL è disponibile anche la classe `RangeImage` che permette, invece, di creare range image per sensori per i quali la superficie di proiezione non è un piano ma un settore sferico. Il metodo `setDepthImage` è utilizzato per costruire l'immagine di range da un vettore di profondità indicando anche la larghezza ed altezza dell'immagine, il centro di essa e le distanze focali normalizzate calcolate come indicato nel paragrafo 3.6.4.

Utilizzando la risoluzione massima del sensore si avranno quindi immagini larghe 1536 pixel ed alte 512 pixel ed il centro dell'immagine sarà esattamente a metà larghezza ed altezza. Le due distanze focali normalizzate dipendono dalla distanza focale del sensore ottico. Il sensore ottico utilizzato nella telecamera disponibile presso il laboratorio di robotica dell'Università di Parma ha una distanza focale di 25 mm, quindi le

distanze focali normalizzate risultano essere quelle calcolate nel paragrafo 3.6.4.

La telecamera *Ranger D50* fornisce l'immagine di prossimità in un sistema di riferimento non collocato sul centro ottico del sensore, come presupposto dagli sviluppatori della libreria *Point Cloud Library*. È stato quindi necessario generare un'immagine rispetto ad un ipotetico punto di osservazione. La relazione fra i due sistemi è semplice e si limita a limitare l'offset fra le due altezze in questo modo:

$$depthVector = Posizione_{sensore} - Profondità_{oggetto} \quad (5.1)$$

Questo vuol dire che si deve semplicemente ricavare il valore del range sul relativo asse dall'informazione di profondità dell'oggetto usando l'equazione 5.1. Nel caso della telecamera con la configurazione geometrica "Reversed Ordinary" l'asse di range coincide con l'asse  $z$ . La posizione del sensore si può prendere ad una distanza arbitraria oppure usare una distanza realmente esistente nel sistema di acquisizione come la *working distance*. In queste applicazioni si è scelto di prendere una distanza arbitraria di 600 mm. Una volta calcolato il vettore di profondità allora si può chiamare il metodo della PCL *setDepthImage* passandogli oltre al vettore, la dimensione dell'immagine, la posizione del centro e le lunghezze focali normalizzate calcolate come nel paragrafo 3.6.4.

Un esempio di immagine di range creata in questo modo è visibile in figura 5.1.

### 5.3 Le caratteristiche NARF

Come già accennato per implementare il riconoscitore di oggetti si vogliono usare delle caratteristiche NARF (Normal Aligned Radius Feature) descritte in [17] ed in [16]. In questo paragrafo si vogliono descrivere brevemente i passi fondamentali dell'algoritmo usato per estrarre le caratteristiche NARF. Per prima cosa nell'algoritmo vi è

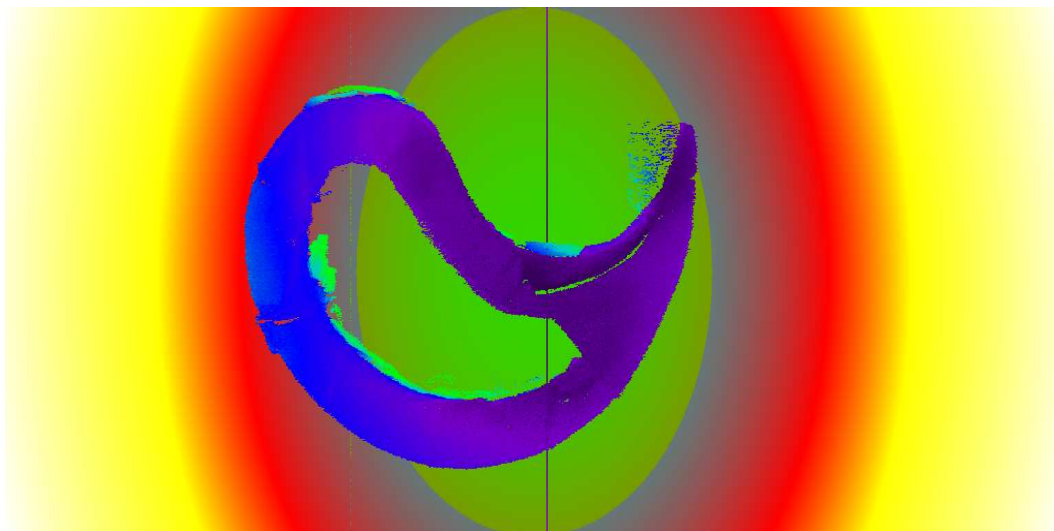


Figura 5.1: Esempio di range image. L'oggetto nell'immagine è un fermaporta(l'oggetto originale è visibile in figura 5.2)



Figura 5.2: Fermaporta

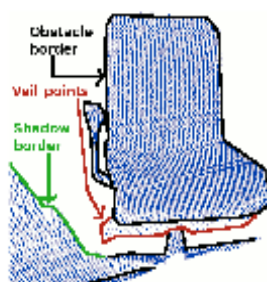


Figura 5.3: Diversi tipi di bordo in un'immagine di range. Immagine tratta da [17]

l'estrazione del bordo e vi è la differenziazione in tre tipi possibili di bordo (figura 5.3):

- **Bordo dell'oggetto:** sono i punti esterni ancora appartenenti all'oggetto;
- **Bordo dell'ombra:** sono i punti del background relativi all'ombra dell'oggetto;
- **Punti velati:** sono i punti di interpolazione fra il bordo dell'oggetto e l'ombra dello stesso.

Con il termine bordo viene inteso non solo il contorno dell'oggetto, ma ogni cambiamento di profondità rilevante presente nell'oggetto. Trovare i cambiamenti di profondità serve ad identificare le zone in cui andare a cercare i punti caratteristici di un oggetto, cioè quelli che servono per la descrizione di un oggetto.

Il secondo passo è quello che permette di ridurre ulteriormente l'area di ricerca dei punti caratteristici, ossia la ricerca dei *punti chiave* dell'oggetto. Questi punti devono avere le seguenti caratteristiche:

- devono avere delle informazioni riguardo al bordo e alla superficie su cui si ricercano;
- devono essere posizionati in punti che possono essere trovati anche se l'oggetto è osservato da un'altra prospettiva;

- devono essere posizionati in un area in cui è possibile eseguire una stima della normale e delle caratteristiche dei punti in generale.

Per estrarre dei punti chiave stabili bisogna avere dei cambiamenti sostanziali della superficie nel vicinato di osservazione, per essere estratti nella stessa posizione anche se osservati da prospettive differenti. Questo significa che ci sono diverse direzioni dominanti dei cambiamenti nell'area esaminata.

Per capire questo bisogna:

- guardare il vicinato di ogni punto immagine e dare un punteggio che indichi il livello di cambiamento della superficie presente nel vicinato. Inoltre viene estratta la direzione dominante del cambiamento ed incluse le informazioni del bordo;
- vengono osservate le direzioni dominanti nei dintorni di ogni punto dell'immagine e viene calcolato un valore che rappresenta: i) di quanto le direzioni differiscono dalle altre e ii) di quanto la superficie nel punto cambia (questo indica la stabilità);
- viene effettuato un livellamento sui valori di interesse;
- non viene ancora effettuata la massima soppressione per trovare i punti caratteristici finali.

Il parametro più importante in questa fase è il *support size*  $\sigma$ , che definisce il diametro della sfera su cui verrà definito e considerato il vicinato di un punto.

L'ultimo passo dell'algoritmo è la ricerca dei punti caratteristici finali dell'immagine (che denominiamo *punti NARF*).

Questi punti hanno il compito di descrivere l'area intorno al punto chiave trovato al punto precedente dell'algoritmo. Le motivazioni che hanno spinto lo sviluppo dei punti NARF sono:

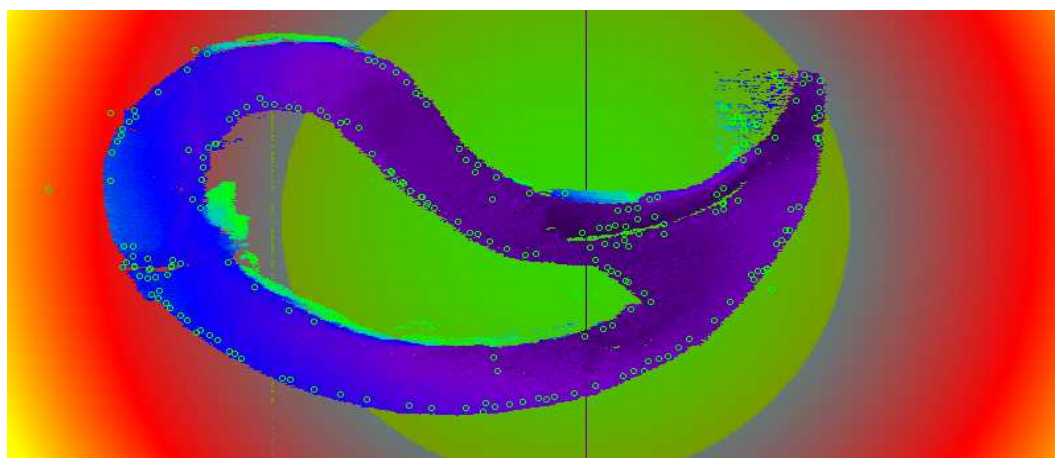


Figura 5.4: Esempio di range image con individuazione dei punti chiave

- capire lo spazio immagine occupato e lo spazio libero, in modo tale da descrivere sia la superficie dell'oggetto, sia la sua forma esterna;
- robustezza rispetto al rumore che può colpire la posizione del punto chiave;
- permettere l'estrazione di un unico riferimento locale del punto di interesse.

Una volta trovati i punti NARF, l'algoritmo restituisce sia le coordinate  $(x,y,z)$  del punto, sia il suo orientamento attraverso gli angoli di *roll*, *pitch* e *yaw*. Infine viene restituito un vettore dei descrittori del punto di 36 elementi. Un esempio di range image con l'individuazione dei punti chiave è visibile in figura 5.4. Come si può notare dall'immagine i punti chiave vengono trovati tutti vicino ai cambiamenti di profondità. Si può anche notare che a causa di rumore o imperfezioni dell'oggetto alcuni punti vengono individuati al di fuori dell'oggetto stesso e questo può rappresentare un problema perchè potrebbe falsare il confronto con altre immagini.

Il problema viene risolto come descritto nel paragrafo 5.4, cioè estraendo il cluster di punti appartenenti all'oggetto e prendendo in considerazione solo i punti NARF appartenenti al cluster.

## 5.4 Clusterizzazione e selezione dei punti caratteristici

Per ovviare al problema del rumore e dei punti caratteristici trovati all'esterno dell'oggetto si è pensato di estrarre il cluster dei punti appartenenti all'oggetto e prendere in considerazione solo i punti NARF contenuti nel cluster.

### 5.4.1 Estrazione del cluster

Si opera nelle ipotesi che l'oggetto sia appoggiato su un piano e che nella scena osservata sia presente un solo oggetto oppure più oggetti ma sufficientemente distanti fra loro. Per svolgere questa operazione è stata definita una classe `Cluster` con il compito di estrarre il cluster dei punti appartenenti all'oggetto direttamente da una *PointCloud*. Passando alla classe la cloud di input ed una di output, la classe restituisce quest'ultima con il risultato della clusterizzazione.

Per estrarre il cluster le operazioni che si effettuano nella classe sono le seguenti:

- primo filtraggio per ridurre il numero dei punti della cloud attraverso l'uso del filtro `VoxelGrid`, implementato nella libreria PCL;
- ricerca ed eliminazione del piano dominante, attraverso la classe `clSACSegmentation` della PCL;
- estrazione dei cluster attraverso la classe `EuclideanClusterExtraction`, sempre implementata nella PCL.

Il risultato di questa operazione è una lista di cluster ordinati per numero di punti. Nelle applicazioni si è sempre lavorato con un singolo oggetto nella scena, quindi per estrarre il cluster dei punti appartenenti all'oggetto basta prendere il primo cluster risultante dall'algoritmo, cioè quello con il maggior numero di punti.

Nel caso ci fossero più oggetti nella scena, allora bisognerebbe modificare l'algoritmo

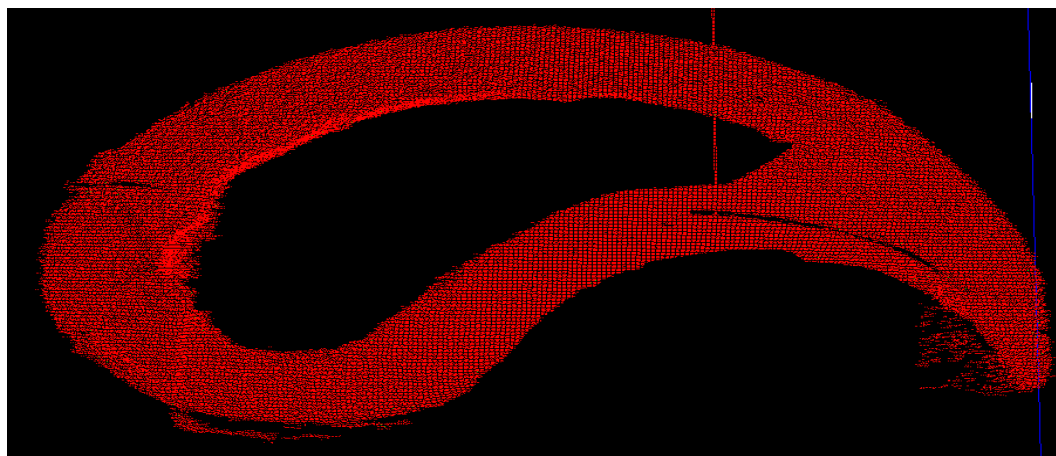


Figura 5.5: Esempio di estrazione del cluster

inserendo, dopo l'identificazione dei cluster, una selezione in base al numero dei punti per identificare quanti oggetti sono presenti nell'immagine. Un esempio di cluster è mostrato in figura 5.5.

#### 5.4.2 Selezione dei punti caratteristici

Ora che si ha il cluster dei punti appartenenti all'oggetto vi è il problema di prendere solo i punti caratteristici appartenenti all'oggetto, quindi quelli contenuti nel cluster. La cosa più banale a cui si può pensare è usare un bounding box, ma ovviamente questo risulta poco preciso, poichè non è aderente al contorno dell'oggetto. Questo vorrebbe dire non risolvere il problema e considerare anche i punti caratteristici non giacenti sull'oggetto. Si è pensato allora di usare l'*Octree*, una struttura dati ad albero per la suddivisione dello spazio tridimensionale, che consente di etichettare le regioni dello spazio occupate dai punti del cluster (figura 5.6). L'*Octree* è molto diffuso per la suddivisione ricorsiva dello spazio tridimensionale ed è già implementato nella libreria PCL attraverso la classe *OctreePointCloud* [12]. Un parametro molto importante della classe è la risoluzione dell'*octree* (*resolution*); questo parametro va settato in base alla grandezza dell'oggetto. Nel caso di oggetti molto piccoli, come nel

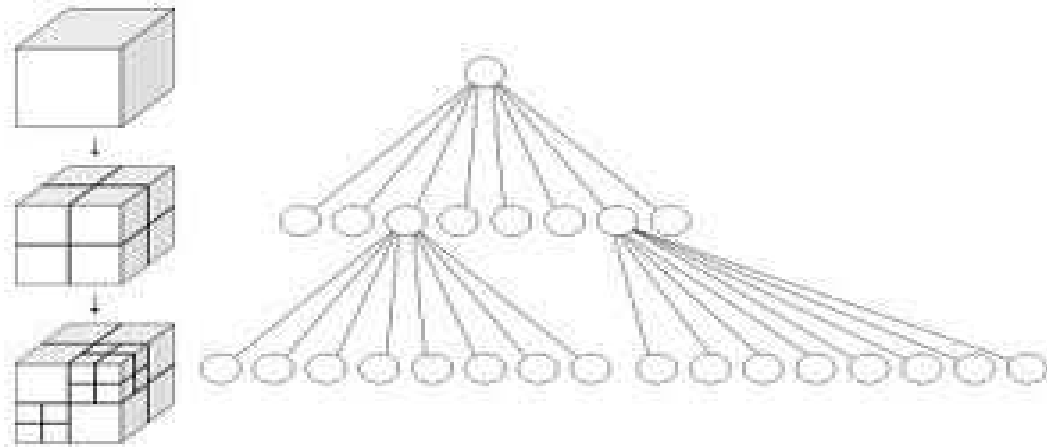


Figura 5.6: Suddivisione ricorsiva dello spazio tridimensionale con l'Octree

caso delle applicazioni descritte, si deve settare il parametro ad un valore basso (per esempio 1 mm) per avere una suddivisione minuziosa dello spazio. Creato l'oggetto della classe `OctreePointCloud`, insieme alla sua risoluzione, bisogna settare la cloud di input, ossia lo spazio tridimensionale che si vuole dividere, che in questo caso sarebbe la nuvola di punti del cluster, attraverso la funzione `setInputCloud`. Data in input la cloud di ingresso si deve creare l'octree chiamando la funzione `addPointsFromInputCloud`. Per vedere se un punto appartiene ad una cella creata dall'octree, vi è una funzione (`isVoxelOccupiedAtPoint`) che riceve in ingresso un punto e restituisce un valore booleano: vero se il punto è presente, falso se il punto non c'è. Si riesce così a sapere quali punti caratteristici dell'immagine appartengono all'octree.

## 5.5 Riconoscimento di oggetti

Si pone ora il problema di sviluppare un riconoscitore di oggetti utilizzando le caratteristiche NARF per il set di oggetti mostrato in figura 5.7. Fino ad ora si è riusciti a creare la range image, estrarre le sue caratteristiche e selezionare quelle che appartengono al cluster dell'oggetto in modo tale da evitare il rumore e le caratteristiche



Figura 5.7: Oggetti utilizzati per il riconoscitore: un pennello, un mattoncino *Lego*, una molletta, un fermaporta ed un connettore seriale per PC

che cadono all'esterno dell'oggetto. Ora si deve trovare un algoritmo che permetta di riconoscere un oggetto anche se viene osservato con un altro orientamento. Le caratteristiche NARF, per quanto detto nel paragrafo 5.3 sembrano essere idonee a questa applicazione, poichè in [17] si dice che siano invarianti rispetto ai cambi di prospettiva. Un primo approccio per confrontare i due oggetti (l'oggetto campione che chiameremo *Target* e l'oggetto che si vuole confrontare con esso che chiameremo *Input*) è consistito semplicemente nell'estrazione delle caratteristiche NARF da entrambi e alla loro associazione. L'associazione è stata realizzata attraverso l'algoritmo ungherese, la cui matrice di costo è calcolata usando come metrica la distanza fra i descrittori dei due oggetti (calcolata attraverso la norma di tipo Manhattan). Una volta associate le coppie dei punti, si è calcolata la somma delle loro distanze euclidee. Se tale somma è minore di una determinata soglia, allora i due oggetti coincidono, in caso contrario no. Questo è stato un primo semplice tentativo che però non ha portato a risultati soddisfacenti, facendo intuire che la semplice distanza fra le caratteristiche non basta a descrivere correttamente un oggetto.

Dopodichè si sono sviluppate sostanzialmente due metriche per effettuare il riconoscimento degli oggetti, e sono state effettuate tre prove, due usando la prima metrica ed una usando la seconda metrica. Prima di descrivere separatamente le due metriche si vuole parlare di un'operazione comune a tutte e due gli algoritmi, ossia l'allineamento.

### 5.5.1 Allineamento

Questa operazione consente di trovare una matrice di trasformazione (ossia una traslazione ed una rotazione), definita nello spazio tridimensionale, che permette di allineare più o meno precisamente due nuvole di punti, qualsiasi cosa esse rappresentino.

Per effettuare questa operazione è già definita nella libreria PCL una classe che permette di allineare due point cloud usando oltre alle coordinate dei punti che contengono anche i loro punti caratteristici.

Questa classe è la `SampleConsensusInitialAlignment` ([2]) ed in questo caso vengono usate delle nuvole di punti di tipo `PointXYZ` insieme con delle caratteristiche di tipo NARF. Si può vedere un esempio in figura 5.8 dove vi è l'allineamento di un fermaporta utilizzando come cloud le coordinate  $(x, y, z)$  delle caratteristiche NARF estratte e come *features* le caratteristiche NARF stesse. L'allineamento risulta essere molto dipendente dai parametri usati per definirlo, quindi si deve prestare molta cura alla loro definizione.

Le principali funzioni da usare per impostare i relativi parametri sono:

- **setMaxCorrespondenceDistance(double max):** si passa un valore *max* che indica la massima distanza accettabile durante la fase di allineamento fra la cloud di riferimento (*Target*) e la cloud di input;
- **setMaximumIterations(int numero):** con il valore *numero* si indica il massimo numero di iterazioni consentite all'algoritmo per cercare di allineare le nuvole di punti. Dopo il numero definito di iterazioni l'algoritmo restituirà la matrice

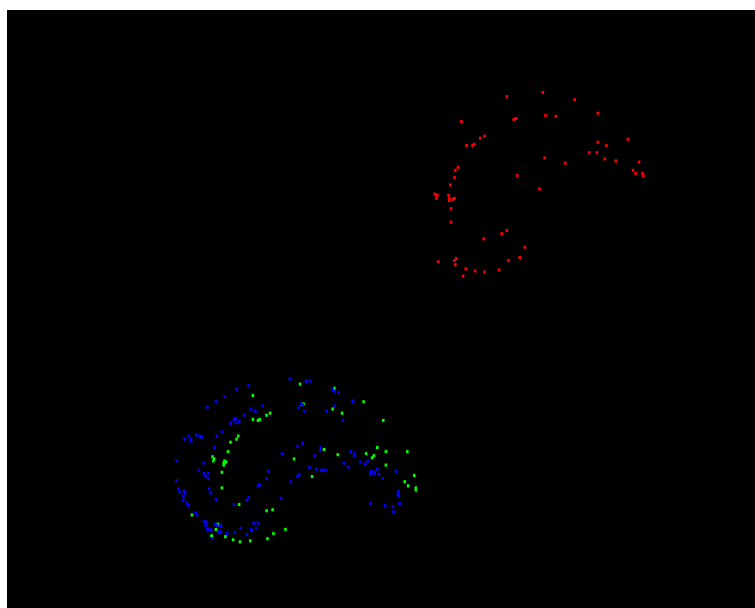


Figura 5.8: Allineamento fermaporta. In rosso si vede la nuvola di punti di input, in blu quella di riferimento ed in verde quella allineata

di trasformazione che definisce il miglior allineamento. Queste sono le funzioni sulle quali si è visto che un cambiamento, anche se minimo, può portare a seconda dei casi, ad un degrado o ad un miglioramento sostanziale delle prestazioni. Vi sono però altre funzioni che possono essere utili per il miglioramento delle prestazioni; se ne citano solo alcune:

- **setRANSACOutlierRejectionThreshold(double soglia):** con *soglia* si definisce un valore massimo per l'algoritmo RANSAC ([4], [14]) interno all'allineamento. L'algoritmo considera che i punti siano degli inliers, ossia che appartengono alla nuvola di punti, se la distanza fra la cloud di riferimento e quella trasformata è minore di *soglia*. Il valore è espresso in metri.
- **setTransformationEpsilon(double epsilon):** setta l'epsilon di trasformazione, ossia la massima differenza permessa fra due trasformazioni consecutive.

### 5.5.2 Classificazione in base all'errore di allineamento

Il primo criterio a cui si è pensato è stato quello di prendere in ingresso le due *Point Cloud* degli oggetti da confrontare, quindi una di input ed una di riferimento e dopo un allineamento iniziale dei due insiemi di punti, è stata effettuata una fase di associazione dei punti in base alla distanza euclidea. Dopo la fase di associazione viene calcolata la distanza fra le due *Point Cloud* come somma delle distanze euclidee delle coppie di punti associate. Tale somma viene utilizzata come termine di confronto per la classificazione e si può considerare come un errore di allineamento. Un oggetto, quindi, viene confrontato con tutti gli altri oggetti campione presenti nel dataset e viene classificato in base all'errore di allineamento. Con questa metrica sono state effettuate due prove, una in cui vengono allineati i cluster dei punti degli oggetti ed una seconda in cui si allineano le coordinate spaziali delle caratteristiche NARF estratte. Le matrici di confusione di queste prove sono visibili nel paragrafo 5.6, nella matrice 5.1 per la prova con l'allineamento del cluster e nella matrice 5.2 per la prova con l'allineamento delle coordinate spaziali delle caratteristiche NARF.

Per la prima prova è stato utilizzato un numero di iterazioni massimo per l'allineamento di 100 visto che la mole di punti presente nei cluster è considerevole (nell'ordine delle decine di migliaia) mentre per l'allineamento delle caratteristiche NARF si sono rese necessarie 10000 iterazioni. Per entrambe le prove è stata impostata una distanza di corrispondenza massima di 0,2.

### 5.5.3 Classificazione in base alla percentuale di matching

Il secondo criterio presuppone, come nel primo, una fase di allineamento. La differenza con il primo metodo è nella fase di associazione e confronto.

A differenza del primo metodo, che associa i punti in base alla distanza euclidea fra cloud di riferimento e cloud di ingresso, questo criterio li associa, usando l'algoritmo

ungherese, in base alla distanza nello spazio dei descrittori delle caratteristiche NARF. Un secondo passo prevede il calcolo della distanza euclidea fra le coppie di punti associate ed il conteggio del numero di coppie che hanno una distanza minore di una certa soglia. L'oggetto di ingresso viene classificato in base alla percentuale di coppie associate che hanno una distanza inferiore ad una certa soglia.

La matrice di confusione di questa prova è visibile nel paragrafo 5.6 nella matrice 5.1. In questa prova il numero di iterazioni massimo è stato impostato a 10000 con un valore di distanza massima tra l'input ed il target di 0,2.

## 5.6 Risultati

In questo paragrafo vengono riportati i risultati ottenuti nelle varie prove di riconoscimento di oggetti effettuate e descritte nel paragrafo 5.5.

I valori nelle matrici di confusione sono riportati in percentuali di oggetti riconosciuti.

	<b>Lego</b>	<b>Fermaporta</b>	<b>Molletta</b>	<b>Pennello</b>	<b>Connettore</b>
<b>Lego</b>	0	50	12,5	25	12,5
<b>Fermaporta</b>	0	75	12,5	12,5	0
<b>Molletta</b>	0	0	100	0	0
<b>Pennello</b>	0	0	25	75	0
<b>Connettore</b>	0	37,5	0	0	62,5

Tabella 5.1: Prima metrica: matrice di confusione per la prova con allineamento dei cluster

Queste le matrici di confusione per la prima metrica. Si può notare che nella prova con l'allineamento del cluster dei punti degli oggetti, l'elemento con il peggior risultato è il *Lego* che non è mai riconosciuto. La causa di questo può essere trovata nel fatto che il lego ha una forma molto comune e quindi viene allineato molto bene con gli altri oggetti con cui viene confrontato. Questo infatti non può accadere con la *Molletta* o

	<b>Lego</b>	<b>Fermaporta</b>	<b>Molletta</b>	<b>Pennello</b>	<b>Connettore</b>
<b>Lego</b>	50	25	12,5	12,5	0
<b>Fermaporta</b>	0	100	0	0	0
<b>Molletta</b>	0	37,5	37,5	25	0
<b>Pennello</b>	0	0	0	100	0
<b>Connettore</b>	12,5	0	37,5	37,5	12,5

Tabella 5.2: Prima metrica: matrice di confusione per la prova con l'allineamento delle caratteristiche NARF

con il *Pennello* che infatti hanno percentuali di riconoscimento molto alte.

Nella seconda prova invece, con l'utilizzo delle sole caratteristiche NARF, si ha un riconoscimento ottimale per gli oggetti che hanno molti punti caratteristici (circa 400 per il *Pennello* e circa 200 per il *Fermaporta*) mentre un riconoscimento discreto per quanto riguarda gli oggetti con pochi punti caratteristici.

	<b>Lego</b>	<b>Fermaporta</b>	<b>Molletta</b>	<b>Pennello</b>	<b>Connettore</b>
<b>Lego</b>	87,5	0	12,5	0	0
<b>Fermaporta</b>	12,5	50	12,5	12,5	12,5
<b>Molletta</b>	50	0	50	0	0
<b>Pennello</b>	0	12,5	0	87,5	0
<b>Connettore</b>	62,5	0	12,5	0	25

Tabella 5.3: Seconda metrica: matrice di confusione per la prova con allineamento e associazione in base a distanza fra caratteristiche ed euclidea

Questo secondo criterio sembra essere quello che restituisce il risultato migliore per la maggior parte degli oggetti. Questo si verifica perchè questo metodo unisce un confronto sia fra le coordinate cartesiane dei punti delle *Point Cloud* degli oggetti, sia un confronto nello spazio dei descrittori delle loro caratteristiche NARF.

Indipendentemente dal metodo utilizzato, comunque, vi è una forte dipendenza dei risultati dalla fase di allineamento e quindi dai parametri dell'allineamento stesso. I parametri dell'allineamento, infatti, cambiano in base alla prova che è stata effettuata ed al tipo di dati utilizzati.

# Conclusioni

Questa tesi si è occupata di studiare ed utilizzare la telecamera di profondità *Sick Ranger D50*. La telecamera *Ranger D50* offre prestazioni di grande qualità, consentendo l'acquisizione di immagini con risoluzione  $1536 \times 512$  pixel con un ridotto errore nella misura di range di ciascun pixel. I principali contributi della tesi consistono nella realizzazione di un sistema di acquisizione di immagini di profondità che integra un sensore complesso e potente come il *Ranger D50* e nell'elaborazione delle immagini di profondità finalizzata al riconoscimento di oggetti. Ciascun contributo corrisponde ad una delle due fasi nelle quali si è articolata l'attività.

La prima fase ha riguardato la realizzazione di un apparato sperimentale per l'acquisizione di immagini di prossimità. È stato necessario, inoltre, costruire un calibro per la stima dei parametri intrinseci ed estrinseci della telecamera. È stata sviluppata un'applicazione per l'acquisizione di immagini di prossimità. Nel realizzare il sistema di acquisizione è stato necessario uno studio dei parametri geometrici del supporto, come la distanza dal piano di lavoro della telecamera e della posizione ed orientamento del sistema illuminatore laser-telecamera, che permettono di soddisfare le specifiche richieste. Per questo motivo è stato realizzato un apparato di acquisizione che permette di variare ragionevolmente la configurazione dei vari elementi costitutivi, in modo tale da lasciare completa libertà al momento dell'acquisizione. Un altro aspetto che ha richiesto attenzione è stato il processo di calibrazione della telecamera di profondità. Sono state scelte le caratteristiche dello strumento di calibrazione, come ad esempio

dimensioni, materiale, colorazione e pattern (a dente di sega), in modo tale da poter garantire le massime prestazioni in precisione ed accuratezza che può offrire la telecamera. Il programma per l'acquisizione dell'immagine di profondità impiega ed integra le API messe a disposizione dalla casa costruttrice. Un problema che si è dovuto affrontare è stato quello di acquisire l'immagine senza l'ausilio di un encoder per la determinazione esatta dello spostamento dell'oggetto rispetto alla telecamera e all'illuminatore laser. Oltre alla scelta dell'intervallo di campionamento si è tenuto conto della geometria del sensore per mettere in relazione gli elementi dell'immagine con le relative coordinate cartesiane nel sistema di riferimento mondo. Il risultato è stato soddisfacente e dalle misure effettuate si evince che l'errore statistico della misura di profondità è di circa 0,5 mm, mentre è di circa 1 mm l'errore radiale.

La seconda fase, invece, ha riguardato la valutazione delle prestazioni del sensore al di là delle misure di ripetibilità e precisione ma integrando i dati sensoriali acquisiti tramite la Ranger D50 in un'applicazione più complessa. È stata pertanto sviluppata un'applicazione per il riconoscimento di oggetti sulla base di immagini di profondità. Sono stati utilizzati ed adattati per l'uso con il sensore algoritmi per l'estrazione di punti caratteristici cui è associato un vettore di valori detti *descrittori* potenzialmente in grado di identificarlo in modo univoco o almeno in modo stabile. Recentemente è stato proposto in letteratura il Normal Aligned Radial Feature (NARF), un nuovo tipo di punto caratteristico concepito per individuare elementi stabili nelle immagini di profondità. Il dato sensoriale tridimensionale è stato elaborato con il ricorso ai metodi della libreria open source PCL (Point Cloud Library). Il confronto fra i dati sensoriali acquisiti è stato effettuato sostanzialmente secondo tre approcci, che combinano l'allineamento tra i cluster, l'allineamento tra le coordinate cartesiane delle caratteristiche NARF ed il calcolo di due distanze: la distanza fra gli oggetti allineati e la distanza fra le caratteristiche NARF degli oggetti. L'allineamento consiste nella stima della rototraslazione che minimizza la distanza tra gli insiemi di punti, siano essi

dei cluster o punti delle caratteristiche NARF. L'approccio migliore è stato quello in cui si è realizzato l'allineamento usando i punti caratteristici dell'oggetto, associando poi attraverso la distanza dei descrittori. Dopo l'associazione viene calcolata una percentuale di *match*, andando a determinare quanti punti sono correttamente associati. I risultati della classificazione di oggetti sono soddisfacenti (la percentuale di oggetti riconosciuti si attesta intorno al 60%) se confrontati con la semplicità dei criteri utilizzati, e danno conferma della qualità non solo dell'immagine di range ma anche della precisione ed accuratezza della telecamera *Ranger D50*.

Uno sviluppo futuro di questo lavoro sarà quello di migliorare l'algoritmo di classificazione. Un altro progetto sarà quello di utilizzare la telecamera come strumento di visione per la manipolazione di oggetti, utilizzandola insieme ad un attuatore che permetterà di ampliare le applicazioni di utilizzo della telecamera. Questo progetto comporterà diverse problematiche, tra cui la principale riguarda l'installazione sull'attuatore e la calibrazione della telecamera.

# Bibliografia

- [1] M. Agrawal e K. Konolige. Real-time Localization in Outdoor Environments using Stereo Vision and Inexpensive GPS. In *The 18th International Conference on Pattern Recognition*. 2006.
- [2] N. Blodow, R. B. Rusu, Z. C. Marton e M. Beetz. Partial view modeling and validation in 3d laser scans for grasping. In *Proceedings of the 9th IEEE-RAS International Conference on Humanoid Robots (Humanoids)*. Paris, France, December 7-10 2009.
- [3] F. Chiabrando, D. Piatti e F. Rinaudo. Camere TOF per il rilevamento metrico: test e prime applicazioni. In *Atti 14<sup>a</sup> Conferenza Nazionale ASITA*. 2010.
- [4] M. A. Fischler e R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, vol. 24, June 1981.
- [5] R. B. Fisher e K. Konolige. *Handbook of Robotics*, cap. 22. Springer, 2008.
- [6] R. I. Hartley e A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, seconda ed., 2004.

- 
- [7] G. Lazzaretti. Sistema di acquisizione superfici basato su robot AlphaCU4 e laser 2D CMOS serie ZS-L da 80 a 15 mm. Relazione per il corso di Robotica 2010/2011 dell'Università di Parma.
- [8] D. G. Lowe. Object recognition from local scale-invariant features. In *Proceedings of the International Conference on Computer Vision*. 1999.
- [9] Mesa Imaging. *Mesa SR4000 Data Sheet*, 2011.
- [10] T. Oggier, F. Lustenberger e N. Blanc. Miniature 3D TOF Camera for Real-Time Imaging. *Springer-Verlag*, 2006.
- [11] K. Ohno, T. Kawahara e S. Tadokoro. Development of 3D Laser Scanner for Measuring Uniform and Dense 3D Shapes of Static Objects in Dynamic Environment. *International Conference on Robotics and Biomimetics*, 2008.
- [12] R. B. Rusu. *Semantic 3D Object Maps for Everyday Manipulation in Human Living Environments*. phd, Technische Universitatet Muenchen, Munich, Germany, 10/2009 2009.
- [13] R. B. Rusu e S. Cousins. 3D is here: Point Cloud Library (PCL). In *IEEE International Conference on Robotics and Automation*. 2011.
- [14] R. Schnabel, R. Wahl e R. Klein. Efficient ransac for point-cloud shape detection. *Computer Graphics Forum*, vol. 26(2):pp. 214–226, 2007.
- [15] Sick AG. *Reference Manual: 3D camera with Gigabit Ethernet (D)*, 2010.
- [16] B. Steder, R. B. Rusu, K. Konolige e W. Burgard. Narf: 3D Range Image Features for Object Recognition. In *23rd IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2010.

- 
- [17] B. Steder, R. B. Rusu, K. Konolige e W. Burgard. Point Feature Extraction on 3D Range Scans Taking into Account Object Boundaries. In *IEEE International Conference on Robotics and Automation*. 2011.
- [18] L. Xia, C.-C. Chen e J. K. Aggarwal. Human Detection Using Depth Information by Kinect. *The University of Texas at Austin - Department of Electrical and Computer Engineering*, 2011.