UNIVERSITÀ DEGLI STUDI DI PADOVA

—

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE
DIPARTIMENTO DI TECNICA E GESTIONE DEI SISTEMI INDUSTRIALI

—

CORSO DI LAUREA MAGISTRALE IN INGEGNERIA DELL'AUTOMAZIONE

RICOSTRUZIONE 3D DI CELLA DI LAVORO ROBOTIZZATA

RELATORE: CH.MO PROF. ING. GIULIO ROSATI

LAUREANDO: FABIO SACCHETTO

ANNO ACCADEMICO 2012-2013

ai miei nonni...

“ Quello che per un uomo è magia, per un altro è ingegneria. Sovrannaturale è una parola inconsistente. ”

Robert Anson Heinlein - Lazarus Long l'Immortale, 1973

Indice

Sommario

L'utilizzo di un robot manipolatore richiede in alcune applicazioni di dover gestire un flusso dati proveniente da una telecamera al fine di rilevare ostacoli o riconoscere un oggetto presente entro lo spazio di lavoro. L'unita di calcolo interfacciata ha dunque il compito di calcolare nuovi punti di via per la movimentazione corretta del robot ed infine rilevare la posizione di un oggetto da prelevare.
Presso i laboratori del **DIMEG** è stato sviluppato un programma munito di interfaccia grafica in grado di interagire con il robot *Adept Quattro* e comandarne i movimenti di perlustrazione dello spazio di lavoro.
In questo lavoro di tesi si è avvalsi del sistema di visione 3D $Microsoft\ Kinect^{TM}$ fissato al robot allo scopo di ricostruire graficamente lo spazio circostante e verificare la presenza di oggetti nel volume della cella.

Introduzione

Nell'automazione moderna i sistemi di visione permettono di uscire dalla classica routine determinata dalle movimentazioni stabilite da rigide leggi matematiche. Grazie ad essi è possibile ricercare forme geometriche negli oggetti inquadrati e calcolarne le coordinate nel piano tridimensionale: applicazioni di noto rilievo sono il *bin picking* e l'*obstacle avoidance*.

In questa tesi si analizzeranno nel **capitolo 1** i diversi sistemi di visione basati su telecamere *RGB*, *IR* e sistemi **TOF** sottolineandone pregi e difetti alle tecnologie attuali. In particolare viene documentato con riguardo il *Microsoft Kinect*, utilizzato nella fase sperimentale.

Il **capitolo 2** prende in considerazione l'unità di movimentazione utilizzata per le scansioni, ovvero un robot parallelo a 4 gradi di libertà. Viene analizzato il suo funzionamento e l'ambiente in cui è collocato per la fse di acquisizione.

Nel **capitolo 3** viene trattato l'argomento del *Bin Picking*. Trattasi la più diffusa applicazione della visione tridimensionale nel campo dell'automazione, viene ritenuto necessario un cenno a tal argomento.

Infine, nel **capitolo 4** si descrive l'attività di laboratorio svolta. Nel periodo di tesi è stato sviluppato un programma con interfaccia grafica atto alla perlustrazione della cella di lavoro e alla simulazione del movimento. Il robot viene munito di sistema di visione tridimensionale ($Microsoft\ Kinect^{TM}$) al fine di creare una ricostruzione grafica dell'interno cella e di una testa di gomma. Verrà analizzato il metodo di ricostruzione utilizzato e confrontato con una ricostruzione grafica realizzata da un scannerizzatore professionale a braccio.

Capitolo 1

Sistemi di visione 3D

In questo capitolo vengono descritti i sistemi utilizzati odiernamente per la visione tridimensionale nel campo dell'automazione. In particolare si fa riferimento ai sistemi di visione stereo, ai sistemi che sfruttano la proprietà di triangolazione della luce strutturata ed ai cosidetti sistemi *time of flight*.

1.1 Visione Stereo

Nella visione di tipo stereoscopica vengono solitamente impiegate due telecamere di tipo standard (RGB o B/N) che inquadrano la stessa scena. Solitamente le telecamere vengono scelte dello stesso modello di costruzione in modo da avere ottica identica e stesso sensore, vengono poi poste in modo che i sensori siano complanari ed allineati e i relativi assi delle ottiche siano paralleli.
La telecamera **L** viene scelta come *reference camera* , mentre **R** viene denominata *target camera*. Ovviamente ognuna di esse ha il suo sistema *world* di riferimento ed il piano bidimensionale del sensore. L'apparecchio **L** avrà coordinate (x_L, y_L, z_L) che indicheremo sistema di riferimento L_{3D} e cosi via per il sistema di riferimento R_{3D} della seconda telecamera. I due dispositivi sono caratterizzati anche da un sistema di riferimento bidimensionale (u_L, v_L) e (u_R, v_R).

Convenzione comune è accento prendere il sistema L_{3D} come riferimento della visione stereo e denotarlo dunque S_{3D}.

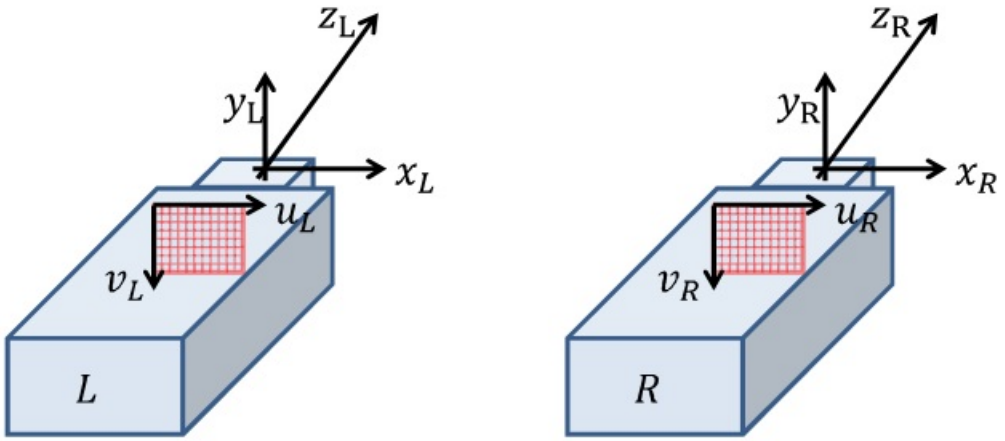


Figura 1.1: coordinate e sistema di riferimento del sistema di visione stereo

Se il sistema è calibrato, un generico punto P con coordinate $\mathbf{P} = [x, y, z]^T$ nel campo immagine di **L** è proiettato al pixel p_L e p_R delle telecamere **L** e **R** con coordinate $pL = [u_L v_L]^T$ e $pR = [u_R = u_L - d, v_R = v_L]^T$. Dalla relazione generata dal triangolo di vertici **P**, p_R e p_L si dimostra che, avendo i punti p_R e p_L le stesse coordinate verticali, la differenza dalle coordinate orizzontali $d = u_L - u_R$, chiamata disparità, è inversamente proporzionale al valore di profondità z di **P**:

$$z = \frac{b|f|}{d}$$

dove b è la distanza tra l'origine di L_{3D} e R_{3D} ed f la lunghezza focale delle due telecamere. I pixel p_R e p_L vengono detti coniugati. Grazie alle coordinate bidimensionali di p_L e la profondità z associata si ottiene le coordinate x e y del corrispondente punto **P** rappresentato rispetto al sistema di riferimento sensore invertendo la matrice di proiezione della telecamera **L**

$$[x, y, z] = K_L^{-1}[u_L, v_L, 1]z$$

Con K_L^{-1} la matrice inversa dei parametri intrinseci di **L**. Quando una coppia di pixel coniugati p_R e p_L sono ammissibili dal sistema, si ottengono le coordinate $\mathbf{P} = [x, y, z]^T$ di un punto, il procedimento attraverso il quale vengono ricavate si chiama triangolazione. La parte più ardua del procedimento sta nel trovare

se ci sono pixel da coniugare, ovviamente parte dei pixel non potranno avere corrispondenza per diversa superficie inquadrata dalle ottiche.

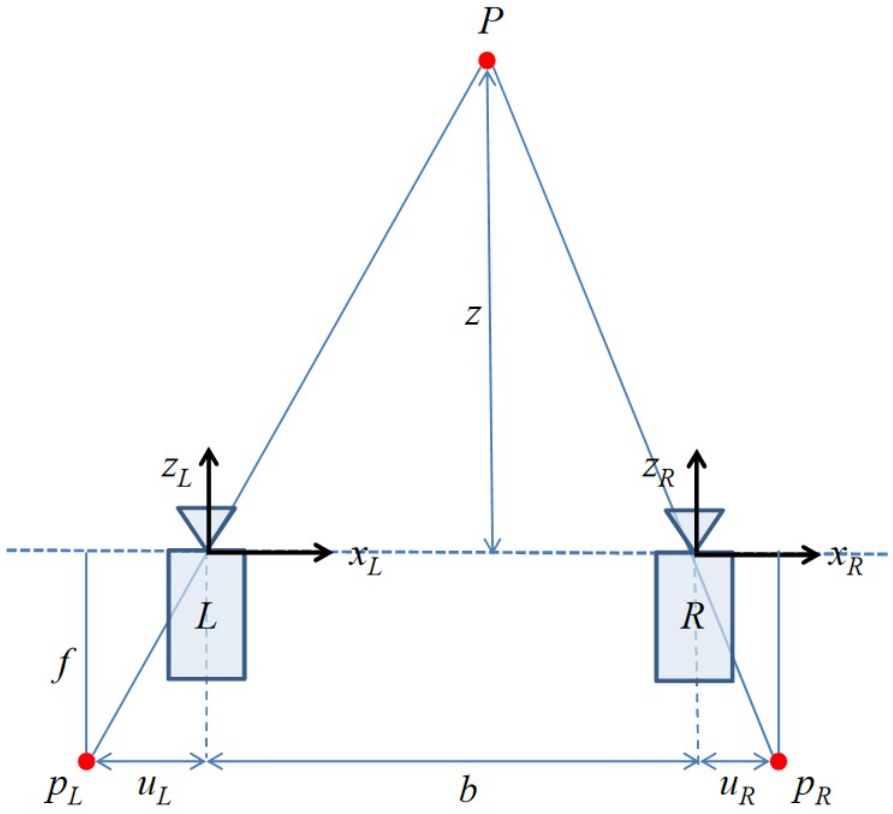


Figura 1.2: coordinate e sistema di riferimento del sistema di visione stereo

Il compito di determinare queste corrispondenze viene affidato ad algoritmi più disparati possibili e riguardano approcci locali oppure globali. I metodi locali considerano solo similarità rinvenute nella zona circostante p_L, andando a verificare se esistono legami con un candidato p_R. Il punto coniugato sarà quello in grado di massimizzare la similarità, per questo il metodo viene chiamato *Winner takes all strategy* [4].

Metodi globali invece non considerano ogni coppia di punti a se stante, ma stimano le disparità una ad una rispettando schemi di ottimizzazione. Essi sono legati alle formulazioni Bayesiane e ricevono tutt'oggi gran riguardo. Queste tecniche modellano i punti di scena minimizzando una funzione costo ed inserendo le correlazioni ad ogni passo in un registro dati che memorizza il costo di ogni

accoppiamento ed il livello di discontinuità trovato nell'immagine.

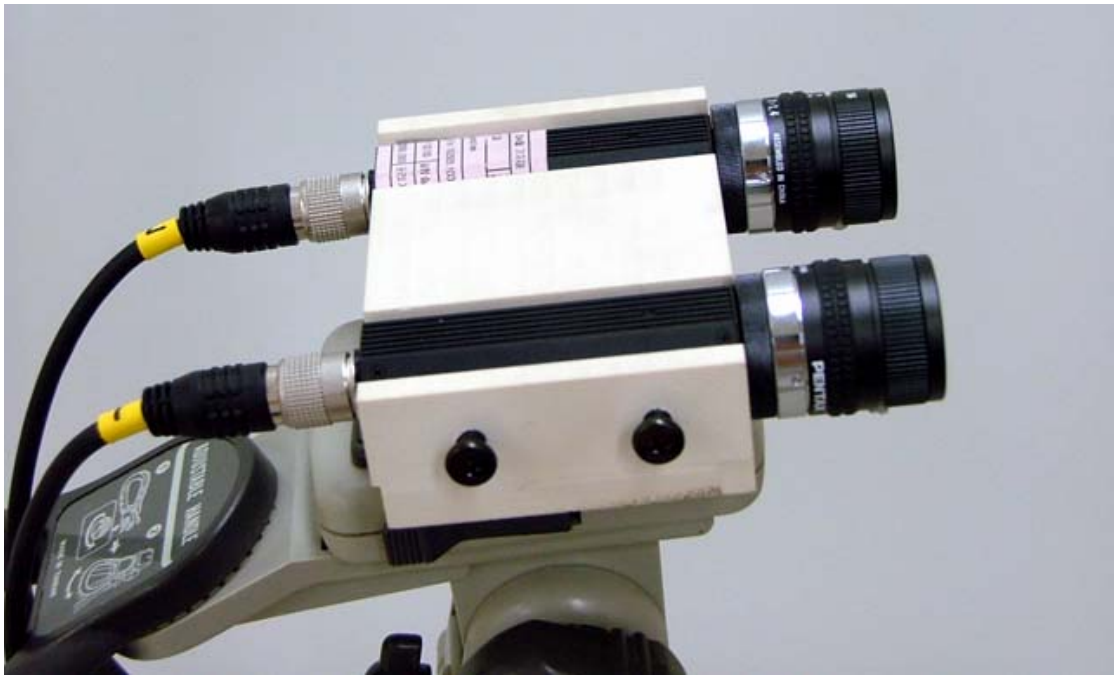


Figura 1.3: coppia di telecamere RGB fissate ad un unico supporto. L'utilizzo di telecamere standard permette di ottenere risoluzione dell?ordine di 2-15 Mpixel

Metodi globali invece non considerano ogni coppia di punti a se stante, ma stimano le disparità una ad una rispettando schemi di ottimizzazione. Essi sono legati alle formulazioni Bayesiane e ricevono tutt'oggi gran riguardo. Queste tecniche modellano i punti di scena minimizzando una funzione costo ed inserendo le correlazioni ad ogni passo in un registro dati che memorizza il costo di ogni accoppiamento ed il livello di discontinuità trovato nell'immagine. La qualità raggiunta dagli ultimi sistemi di ricostruzione 3D stereo dipende tuttavia in maniera inevitabile dalle caratteristiche della scena. Se si pensa infatti a inquadrature di oggetti con colore uniforme e senza particolari geometrie si avranno poche corrispondenze di pixel e di conseguenza si avranno poche informazioni per la ricostruzione tridimensionale della scena inquadrata.

1.2 Telecamere a luce strutturata - Principio triangolazione

Una telecamera a luce strutturata basa il suo principio sull'acquisizione, tramite un sensore a matrice, di una particolare frequenza dello spettro di radiazione luminosa emessa dal proprio emettitore. Queste telecamere sono costituite dunque da due elementi contraddistinti: emettitore e ricettore.

Figura 1.4: Telecamera ShapeDrive serie SD. E' ben visibile il proiettore nella parte destra; la ricezione dell'immagine viene affidata da un'ottica ed un sensore da B/N da 3 Mpixel

L'emettitore usualmente è costituito da un power led ad infrarossi, la luce sviluppata fa parte dunque del range di frequenze al di sotto del limite minimo percettibile all'occhio umano, ma in alcuni casi trova applicazione il laser. Questa luce si dice strutturata poiché tramite un'apposita lente viene direzionata in linee o in griglie, o in casi meno frequenti, in una forma particolare quale una spirale.

Prendiamo a titolo d'esempio una luce indirizzata in linee verticali. Quando una di queste linee viene proiettata su un oggetto, da un punto di osservazione diverso da quello del proiettore viene acquisita l'immagine della linea: essa è deformata rispetto a quella originaria. Attraverso un'elaborazione della linee deformate è possibile risalire alla forma dell'oggetto colpito dalla luce. Il procedimento con cui è possibile ricostruire la forma dell'oggetto colpito è analogo a quello del paragrafo precedente. Mentre nella visione stereo le triangolazione av-

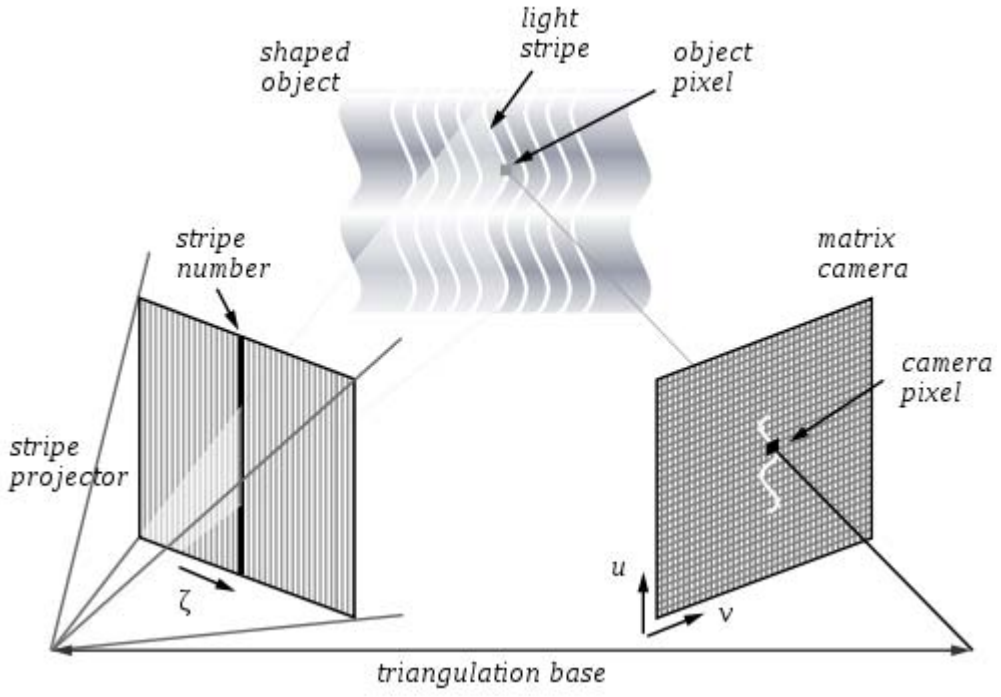


Figura 1.5: triangolazione nelle telecamere a luce strutturata

veniva tra due generici pixel delle telecamere, ora avviene tra un punto preciso della luce strutturata (o da una linea di luce) e un pixel della telecamera.
Questi sistemi sono detti anche a *campo intero* perché per ogni punto sensibile del CCD del sistema di acquisizione si ricava una tripletta (x, y, z). Per tale principio è possibile acquisire ad ogni frame l'intera matrice del sensore ottenendo dunque un'informazione di tutti i pixel rappresentanti la risoluzione della telecamera. Il vantaggio rispetto alla visione stereo consiste proprio nella robustezza dell'acquisizione.

1.3 Telecamere a luce strutturata - Telecamere TOF

Le telecamere TOF (Time of flight - tempo di volo) stanno prendendo piede nel mondo dell'automazione poiché permettono di dare in uscita un elevato flusso di dati e, ancor più importante, le informazioni riguardo la matrice dati acquisita sono espresse come distanza dal centro focale della lente.

Ciò permette di riconoscere facilmente la posizione degli oggetti inquadrati dall'ottica del dispositivo.
L'inconveniente di queste telecamere riguarda la risoluzione dei sensori installati. Per ora sfruttano una matrice CMOS di risoluzione $320x240$ pixel e solo in alcuni casi si trova impiego dello standard VGA ($640x480$ pixel) . Grazie ai continui progressi della microelettronica e della micro-ottica queste telecamere stanno soppiantando il mercato delle telecamere a triangolazione. Basti pensare che nel corso del 2010 la TOF camera a maggior risoluzione sviluppava $204x204$ pixels.

Figura 1.6: Mesa Imaging SR4500

Figura 1.7: Fotonic E-series

Ancora una volta queste telecamere sfruttano il principio dell'emissione di una sorgente luminosa di tipo NIR (*Near Infrared Light* , $700 - 1400um$).
Il tempo di volo, ovvero il tempo impiegato dalla sorgente luminosa per colpire l'oggetto ed arrivare al sensore, viene misurato con varie metodologie. Si distinguono quindi le telecamere a luce pulsata e quelle a modulazione continua dell'onda.

1.3.1 Telecamere ToF a modulazione continua dell'onda

In questo contesto ci focalizzeremo nei sistemi in cui il segnale luminoso segue una funzione periodica del tempo (di tipo sinusoidale). Mentre i sensori basati sulle pulsazione della sorgente luminosa misurano il tempo tra l'invio e la ricezione della luce emessa, i sensori presi in esame calcolano il cosidetto *tempo di volo* semplicemente dalla differenza di fase tra segnale emesso e ricevuto.

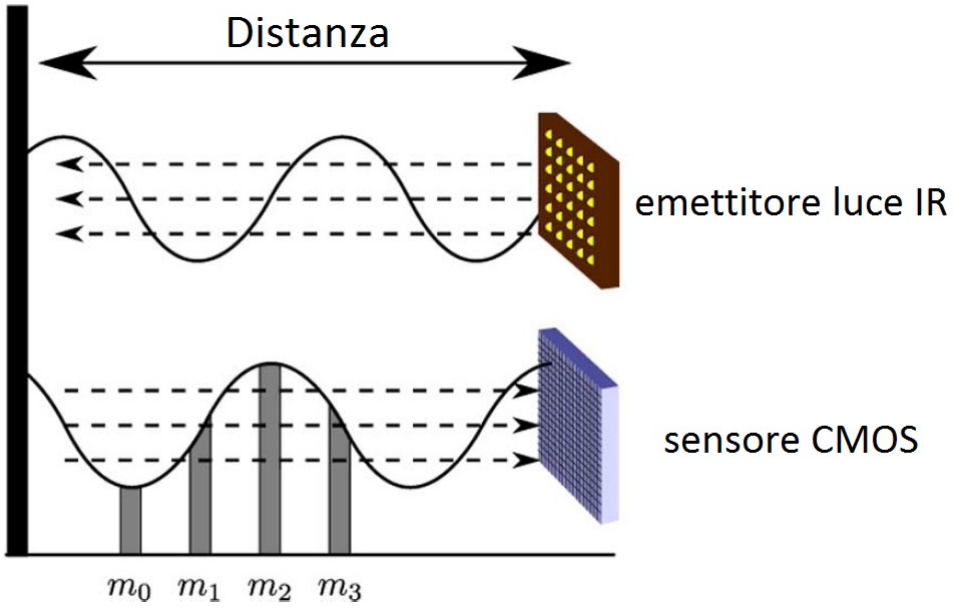


Figura 1.8: misurazione della distanza attraverso la fase dell'onda

Ogni pixel del sensore campiona il segnale ricevuto in forma di fotoni quattro volte per ogni periodo dell'onda (m_0, m_1, m_2, m_3 nella figura 1.8), ottenendo quindi fase

$$\varphi = \arctan(\frac{m_3 - m_1}{m_0 - m_2})$$

offset

$$B = \frac{m_0 + m_1 + m_2 + m_3}{4}$$

e amplitudine

$$A = \frac{(|m_3 - m_1|^2 + |m_0 - m_2|^2)^{\frac{1}{2}}}{2}$$

Questa tecnica viene conosciuta come campionamento *four-bucket* e permette di calcolare facilmente il valore di profondità cercato tramite la formulazione

$$D = L\frac{\varphi}{2\pi}$$

dove L è un parametro intrinseco del sistema, ovvero

$$L = \frac{e}{2f_m}$$

dove e indica la velocità della luce nell'aria, mentre f_m è la frequenza dell'onda luminosa emessa dal led. I valori di intensità (B) a di ampiezza (A) del segnale vengono solitamente utilizzati per predire la qualità delle misurazioni.
In contraddizione alle telecamere TOF che sfruttano la misurazione della fase in maniera analogica,stanno comparendo in commercio le varianti digitali. Performance migliori sono ottenibili infatti grazie alla tecnologia SPSD *(Single-photon synchronous detection)*. Il sensore impiegato non sarà più quindi di natura CCD o CMOS, ma viene costruito da una matrice di diodi a valanga sensibili al singolo fotone (SPSD). Grazie alla natura digitale di questi ultimi non sono presenti errori dovuti alla conversione dall'analogico al digitale. I calcoli poi effettuati per la rilevazione della distanza sono i medesimi della variante analogica.

1.3.2 Telecamere ToF a luce pulsata

Queste telecamere utilizzano il medesimo comparto *hardware* delle altre ToF. Si contraddistinguono solamente nel sistema di gestione della sorgente luminosa e per i calcoli facilitati al fine di rilevare la distanza.
In questo caso viene gestita l'emissione della sorgente luminosa da parte del led-IR in modo da generare fotoni solo quando richiedo. Il tempo di volo (t_f) sarà semplicemente misurato come differenza temporale tra l'emissione e la ricezione del segnale. Con la notazione del paragrafo precedente, la distanza sarà calcolata nel seguente modo:

$$D = \frac{t_f e}{2}$$

Una nota in favore di questi sistemi consiste nell'accuratezza di acquisizione del *tempo di volo*. L'inconveniente di questo metodo consiste nella non emissione continua dell'onda, cioè del segnale, con conseguente diminuzione della frequenza di acquisizione dei fotogrammi.

1.4 Confronto tra i vari sistemi

Nei capitoli precedenti non è stato introdotto volutamente il concetto dei sistemi basati sull'emissione di una luce laser. Questi permettono un'alta qualità dell'immagine in termini di risoluzione ed accuratezza. Tuttavia quando si parla di sensore laser 3D si fa riferimento ad un sistema di scannerizzazione. L'immagine viene creata infatti grazie ad un movimento da parte dell'oggetto o attraverso la movimentazione del fascio laser; ciò comporta a dei tempi di acquisizione rilevanti che compromettono l'utilizzo di questi ultimi in particolari situazioni.
I sistemi basati sulla presenza di almeno due telecamere (o attraverso specchi per riflessione) permettono come spiegato in precedenza di ricostruire dei punti nel campo tridimensionale. Tuttavia essi non sfruttano una luce appositamente emessa, perciò zone troppo illuminate od ombre oltre che a particolari geometrie possono compromettere le corrispondenze tra due pixel dei sensori. Come nelle telecamere a luce strutturata la sensoristica si affida a telecamere CCD o CMOS standard e quindi è possibile ottenere elevati livelli di risoluzione grafica. Da notare che queste due tecnologie necessitano di un comparto software in grado di risolvere il *problema della corrispondenza* con algoritmi che, in alcuni casi, compromettono l'immediatezza di risultati. Essa è ottenibile dunque con le moderne telecamere ToF; grazie all' *hardware* totalmente diverso esse permettono rilevazioni immediate ad alte frequenze a scapito però di risoluzione modeste.

differenze	**ToF**	**stereo visione**	**luce strutturata**
problema corrispondenza	No	Si	Si
calibrazione fattori estrinseci	No	Si	Si
auto-illuminazione	Si	No	Si
supercifi monocromatiche	Buone prestazioni	Pessime prestazioni	Buone prestazioni
range di profondità	0,3 - 7,5 m	dipendente da risoluzione e posizione	dipendente dalla potenza della luce emessa
risoluzione immagini	fino a 640*480	alta risoluzione	alta risoluzione
fotograami/secondo	oltre 25 fps	25 fps	25 fps

Figura 1.9: differenze tra le varie tecnologie impiegate nel campo tridimensionale

1.5 Microsoft Kinect^{TM}

$Kinect^{TM}$ è un dispositivo inizialmente sviluppato da Microsoft per la consolle gaming XBOX 360, successivamente commercializzato anche per i PC.
$Kinect^{TM}$ è munito di telecamera RGB con risoluzione 640x480 pixel, telecamera ad infrarossi 640x480 pixel ed un proiettore laser IR da circa 1,2 Watt. La capacità di acquisizione permette di inviare immagini alla frequenza di 30 *fps*. Inoltre esso dispone anche di un array di microfoni utilizzato dal sistema per la calibrazione dell'ambiente in cui ci si trova, mediante l'analisi della riflessione del suono sulle pareti e sull'arredamento. La barra del $Kinect^{TM}$ è motorizzata lungo l'asse verticale e può eseguire un movimento di 15 gradi in ambi i sensi. Grazie al sensore gravitazionale è possibile porre l'ottica del $Kinect^{TM}$ in posizione orizzontale.

Figura 1.10: $Microsoft\ Kinect^{TM}$ per PC

Allo stato attuale il $Kinect^{TM}$ risulta il prodotto di elettronica di consumo più venduto con oltre 25 milioni di pezzi e risulta già in fase di progettazione il suo successore. Il funzionamento del $Kinect^{TM}$ è complesso poiché vanta al suo interno di un sistema *hardware* di buon livello ed algoritmi per la riduzione del rumore. Il tutto si basa su un chip prodotto dall'azienda israeliana *PrimeSense* installato anche su dispositivi per il gaming o per il campo medico prodotti da aziende concorrenti.

Il dispositivo quindi è più che una semplice telecamera; oltre al movimentazione (*tilt*) al suo interno è alloggiata una scheda madre con relativa CPU, memoria

Figura 1.11: ottica del dispositivo messa in vista

Figura 1.12: particolare del sistema di regolazione dell'inclinazione

RAM e memoria di massa, nonchè sistema di raffreddamento e ventilazione.

Sebbene la commercializzazione del $Kinect^{TM}$ si stata massima ed il suo funzionamento sia stato analizzato da vari centri di ricerca di varie università, non è ben chiaro il principio di funzionamento e il funzionamento software integrato. Si può dedurre tuttavia che il dispositivo utilizza un sistema di illuminazione a luce strutturata e che il calcolo del fattore profondità sia affidato ad algoritmi di triangolazione.

1.5.1 Kinect^{TM}, principio di funzionamento

Il principio utilizzato per il calcolo della profondità va sotto il nome di *triangolazione attiva*. Denominiamo la telecamera IR presente al su interno con **C** ed il proiettore led-IR con **A**. Di conseguenza un punto p_C apparterrà all'immagine acquisita dal sensore, mentre un punto p_A fa parte della matrice di proiezione; la profondità z rilevata dal punto 3D **P** associato a p_C si ottiene nel medesimo modo spiegato in precedenza.
La luce emessa dal led IR è dunque invisibile all'occhio umano e la telecamera sarà tale da acquisire solamente la frequenza d'onda generata, in modo da ovviare a disturbi dovuti a sorgenti luminose esterne.

La presenza della luce solare infatti crea un disturbo all'immagine in quanto il 45 della potenza luminosa emessa fa parte della gamma *infrarossa* . Per ovviare a tal problema i produttori cercano di rendere il sensore CMOS il più selettivo possibile e la luce proiettata, nel caso di telecamere ad uso industriale può arri-

Figura 1.13: disassemblaggio del $Microsoft\ Kinect^{TM}$

vare ad usufruire di una matrice led di 10 Watt. Nel caso del $Kinect^{TM}$ tuttavia il flusso luminoso emesso è modesto e dunque è preferibile l'utilizzo in ambienti chiusi o illuminati artificialmente.
Ritornando al contesto, il flusso dati disponibile dal dispositivo comprende una serie di dati alla frequenza di 30 *fps*:

- l'immagine acquisita da **C**, nel caso specifico è un'immagine IR chiamata I_K definita dal reticolo G_K associato al sensore **C**. Con la notazione già introdotta, gli assi che identificano G_K coincidono con u_C e v_C. Tutti i valori rilevati in pixel hanno valori compresi tra 0 ed 1.

- la cosidetta *disparity map*, ovvero la distanza tra i pixel originari della matrice di proiezione ed i pixel nella matrice del sensore di acquisizione; indicata con $\hat{D}_K$. I valori della matrice sono indicati con $[d_{min}d_{max}]$ dove d_{min} e d_{max}

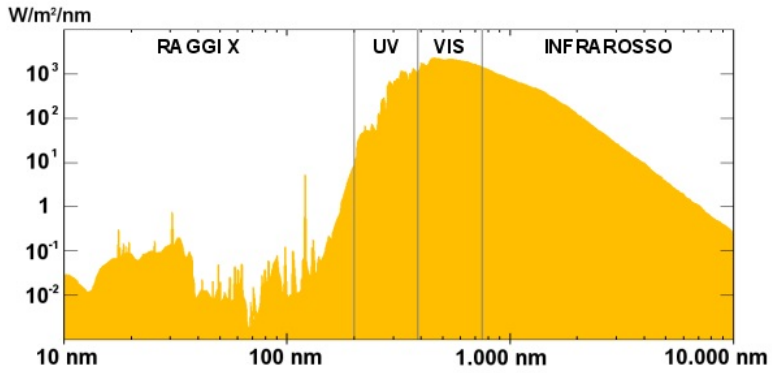


Figura 1.14: spettro della luce solare

sono il massimo ed il minimo permessi dai valori di diparità.

- la matrice *depth*, ovvero l'immagine espressa con i valori di profondità. Essa è il risultato dei calcoli di triangolazione ottenuti a partire dalla matrice $\hat{D}_K$. Nominiamo quest'ultima con $\hat{Z}_K$: è definita nel reticolo G_K del sensore **C** ed assume valori compresi tra $z_{min} = \frac{bf}{d_{max}}$ e $z_{max} = \frac{bf}{d_{min}}$.

- la matrice *RGB*, ovvero una matrice che definisco $\hat{M}_K$ associata a **B** , ovvero il sensore CMOS RGB. Ogni cella di questa matrice comprende i tre valori delle tonalità Rosso, Verde e Blu entro i valori limite 0 ed 1.

La risoluzione delle matrici I_K,$\hat{D}_K$, $\hat{Z}_K$ è di 640x480 pixel. La distanza minima misurabile è di 0.5 m, mentre la massima è di 15 m. I valori sopra citati b e f valgono $75mm$ e $585.6pixel$. La disparità va da un minimo di 2 pixel al massimo di 88 pixel.

Ogni pixel della matrice N_R x N_C I_K viene coinvolto dall'algoritmo di triangolazione. Il $Kinect^{TM}$ infatti applica simultaneamente l'algoritmo a tutti i pixel, per cui il metodo va sotto il nome di *triangolazione matriciale* . La difficoltà di questo metodo sta nella capacità di rendere l'algoritmo semplice come nel caso di un punto singolo. Il problema può essere risolto più agevolmente se il flusso

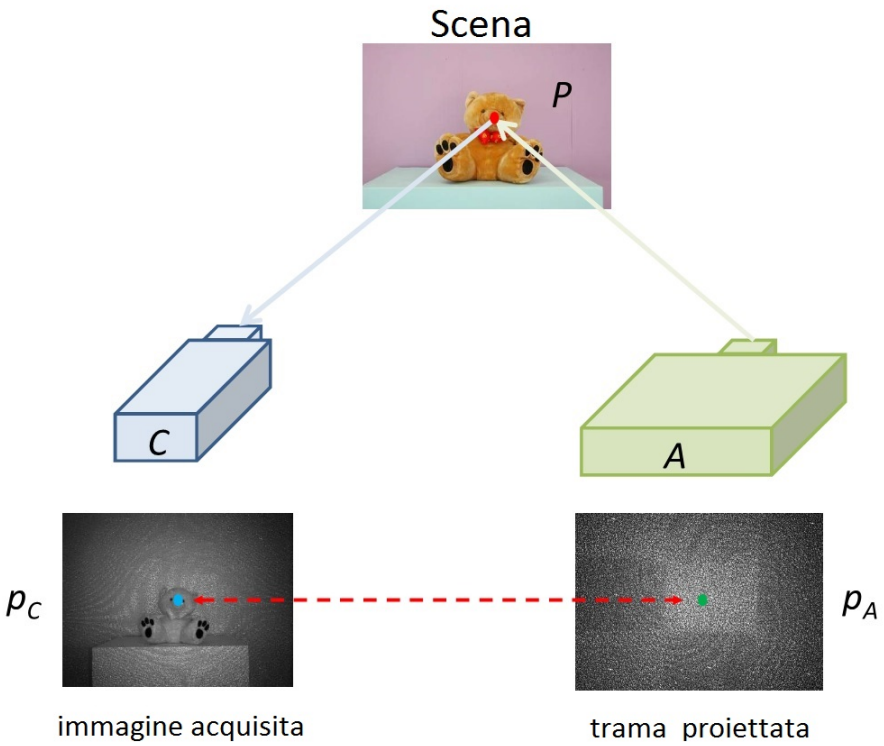


Figura 1.15: Funzionamento della triangolazione: dal punto A parte la proiezione di una funzione grafica sulla scena ed il punto C acquisisce per riflessione. Il coniugato del punto p_A è p_C ed il punto 3D nella scena associato a p_A è **P**.

luminoso proiettano utilizza dei metodi di codifica. Il design utilizzato nella proiezione del fascio IR da parte del $Kinect^{TM}$ rappresenta il suo punto di forza. Prima di accennare al caso specifico del $Kinect^{TM}$ si discuterà nel prossimo paragrafo delle tecniche di codifica della luce.

1.5.2 Metodi di codifica della luce

Assumiamo che il fascio proiettato sia formato da N_R^AxN_C^A pixel dove N_R^A e N_C^A rappresentano il numero di righe e colonne. Per applicare il metodo della triangolazione ogni pixel deve essere associato ad una *parola-codice*, ovvero ad una specifica configurazione locale del modello grafico(o trama) proiettato. Il fascio viene dunque emesso da **A**, viene riflesso dalla scena ed infine catturato da **C**. Un algoritmo di corrispondenza analizza le *parole-codice* nell'immagine acquisita I_K in modo da trovarne il pixel coniugato appartenente al modello grafico proietta-

to. La motivazione dell'utilizzo di questi modelli grafici per la proiezione sta nel fatto di rendere le parole-codice decodificabili anche in presenza di non-idealità del processo di proiezione o di acquisizione.

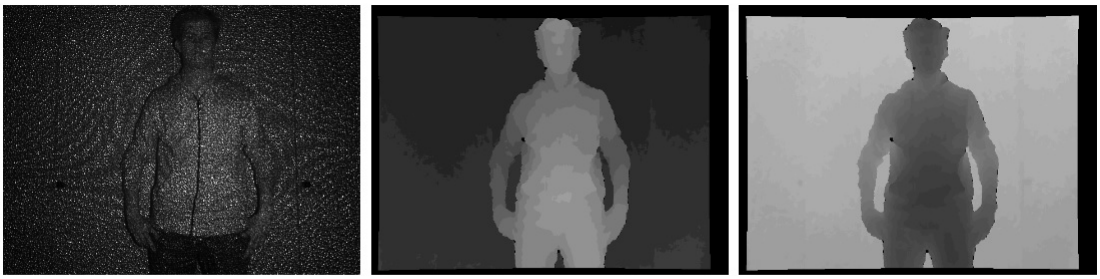


Figura 1.16: Esempio di acquisizione: a) pattern IR acquisito dalla telecamera RGB; b) matrice immagine di disparità; c) immagine di profondità. [4]

Prendiamo in considerazione cosa rende una *parola-codice* decodificabile. E' intuitivo che più le *parola-codice* sono differenti più sarà robusto il sistema dai disturbi e dalle interferenze con gli altri. Data la totalità di elementi che compongono la trama riflessa, minore sarà il quantitativo di *parole-codice* differenti e più sarano accentuate le differenze tra una *parola-codice* e l'altra e dunque il codice avrà maggior robustezza. Invece di coniugare punti spaziali con le linee orizzontali,il problema della decodifica può poi esser formulato indipendente per ogni colonna in modo da rendere più piccola possibile la cardinalità di ogni possibile *parola-codice*. Basti pensare che, per ogni riga della trama proiettata ci sono $N = N_C^A$ pixel p_A^1, p_A^2, ... p_A^N da essere codificate con N *parole-codice* w_1, w_2, ... w_N. Ogni *parola-codice* è caratterizzata da una specifica distribuzione nella trama. Chiaramente, più la distribuzione locale di un pixel differisce dalla distribuzione di un altro pixel appartenente alla stessa colonna, più robusto sarà il codice.
Un alfabeto di *parole-codice* può essere generato da un proiettore di luce, infatti esso può produrre n_P differenti valori di illuminazione (es. $n_P = 2$ se abbiamo un proiettore bianco e nero binario, $n_P = 2^8$ per un proiettore a scala di grigi ad 8 bit, $n_P = 2^{24}$ per un proiettore RGB con ogni canale colore ad 8 bit).
La distribuzione locale nella trama per il pixel p_A è data dai valori d'illuminazione

dei pixel intorno a p_A. Nel caso l'intorno considerato abbia n_W pixel, ci saranno $n_P^{n_W}$ possibili configurazioni della trama. Dall'insieme di tutte queste possibili configurazioni ne devono essere scelte N. Il risultato ottenuto dalle *parole-codice* relative ai pixel proiettati nella trama non è altro che ciò che viene proiettato sulla scena ed acquisito da C.

Richiamiamo ora il fatto che, causa le proprietà geometriche di un sistema calibrato, un pixel p_A della trama, con coordinate $p_A = [u_A, v_A]^T$, è proiettato al punto P della scena, con coordinate $\mathbf{P} = [x, y, z]^T$, e acquisito da C in p_C, con coordinate $p_C = [u_A + d, v_A]^T$. Il processo di proiezione ed acquisizione determina dunque un spostamento laterale d inversamente proporzionale alla profondità z. Il valore d è la quantità determinante nel processo di triangolazione attiva (ovvero con telecamera e proiettore), e necessita di essere stimata in maniera accurata in quanto da essa si ricavano la geometria per trovare il punto scena tridimensionale P. Nell'algoritmo di triangolazione in forma matriciale i valori di disparità d sono contenuti in un array per poi esser allocati nella matrice $\hat{D}_K$ (nel caso del $Kinect^{TM}$).

Nel processo appena descritto ci sono un numero di fattori che influenzano la proiezione o interagiscono nel processo di acquisizione che meritano di essere citati:

- *Distorsione della prospettiva.* Dati dei punti di scena aventi diverso valore di profondità z, può capitare che pixel vicini nella trama proiettata non siano mappati come vicini nei pixel di I_C. In questo caso la distribuzione locale della trama acquisita diventa una versione distorta della distribuzione locale della trama proiettata.

- *Distorsione dei colori o della scala di grigi dovute alla distribuzione dei colori nella scene o alla riflessione dei materiali.* La trama proiettata risente della riflessione (e dell'assorbimento) da parte di alcuni superfici. Il rapporto tra la luce incidente e tangente dipende dalla riflettività della scena, che generalmente è correlata alla distribuzione del colore. Nel caso particolare di luce IR, usata anche nel proiettore del $Kinect^{TM}$, l'apparizione di pixel p_C

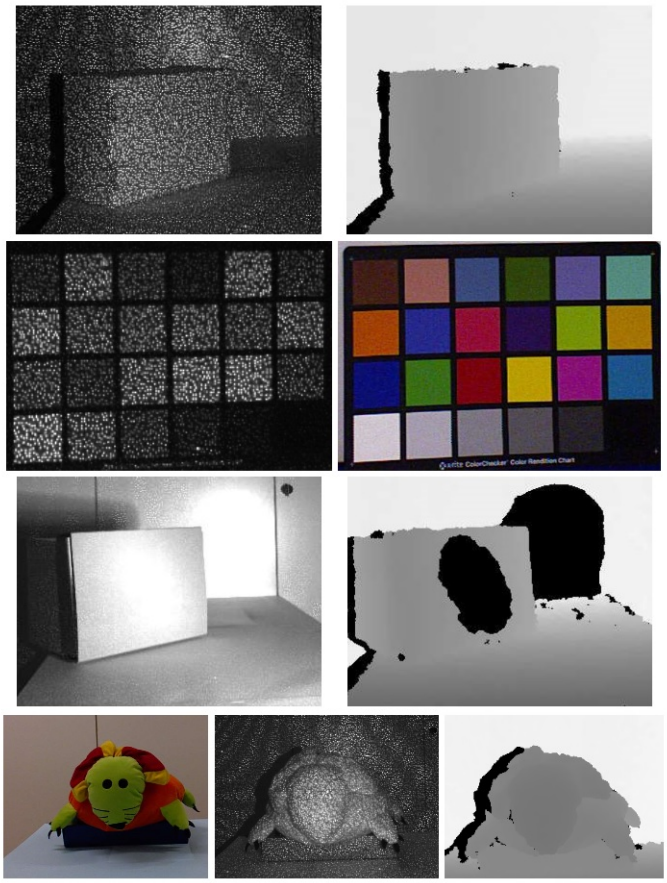
nell'immagine dipende dalla riflettività della superficie inquadrata e dalla frequenza utilizzata per la luce IR. Un pixel di alta intensità in p_A potrebbe indicare uno strano assorbimento dovuto alla bassa riflettività della scena dove viene proiettato, i valori del suo coniugato p_C dunque possono apparire distorti a causa dell'influenza del rumore nel debole segnale luminoso captato.

- *Illuminazione esterna.* Il colore acquisito dalla telecamera RGB non dipende solamente dalla luce ambientale (artificiale o non) che interessa le superfici della scena. Il proiettore genera una frequenza IR (che quindi è percettibile dai sensori RGB), di conseguenza parte del segnale RGB (sopratutto in situazioni di scarsa luminosità ambientale) viene disturbato da esso. Altro fatto da ricordare: l'illuminazione di tipo solare, alogena o al tungsteno comprende le frequenze dell'infrarosso, con conseguente disturbo nell'acquisizione IR.

- *Occlusione.* Causa le differenti posizioni di proiettore e sensore e alla profondità della scena non tutti i pixel che vengono proiettati nella scena vengono visti da C. Causa alcune geometrie della scena non ci può essere fisicamente una corrispondenza tra punti proiettati ed acquisiti. E' importante identificare correttamente i pixel di I_K che non hanno il proprio coniugato in modo da eliminare errate corrispondenze.

- *Non idealità nel proiettore e nella telecamera.* Sia il proiettore che la telecamera non possono essere considerati sistemi ideali causa le non linearità nella proiezione e nell'acquisizione dei colori.

- *Rumore nel proiettore e nella telecamera.* Come tutti i sistemi elettronici, la presenza di rumore Gaussiano influenza la proiezione e l'acquisizione.



Esempi di vari problemi che possono modificare la proiezione; nelle immagini ottenuti dalla matrice di profondità il colore nero indica che non è stato ottenuto una valida misurazione. *Prima riga*: proiezione della trama IR in superfici inclinate e mappa di profondità. Si nota uno delle problematiche del Kinect: si osserva lo shiftamento dovuto al cambio di valori di profondità, inoltre la distorsione della prospettiva compromette le superfici inclinate. *Seconda riga*: il Kinect inquadra una tavola colori. A sinistra la mappa di profondità e a destra. l'apparizione del punto proiettato dipende dal colore e dalla superficie. *Terza riga*: una forte illuminazione esterna influenza la scena acquisita. L'immagine IR satura con conseguente perdita di informazione e corrispondenza tra pixel nella parte interessata. *Quarta riga*: la parte sinistra del pupazzo è ben visibile dalla telecamera RGB e anche dalla telecamera IR. La differente posizione del proiettore rende inaccessibile quest'area dal fascio luminoso generato; la profondità di questa zona non viene rilevata [**?**]

In modo da capire come possono essere inibiti alcuni degli effetti descritti che impediscono di risolvere il problema della corrispondenza è necessario fare due considerazioni. La prima, il processo di corrispondenza è caratterizzato da due decisioni fondamentali:

- quale *parola-codice* viene assegnata ad ogni pixel p_A della trama proiettata. La *parola-codice* corrisponde ad una trama che viene proiettata dal suo centro in p_A (tutte le trame dovute ai pixel vicini sono fuse insieme in una singola proiezione);

- quale *parola-codice* viene assegnata ad ogni pixel p_C di I_K, o equivalentemente, come trovare la *parola-codice* più simile alla distribuzione della trama locale in modo da identificare correttamente il pixel coniugato p_C di p_A.

Una seconda considerazione va fatta in base ai schemi di codifica:

- *Codifica diretta*: la *parola-codice* associata a ciascun pixel p_A è rappresentata da un valore del pixel stesso, come per esempio il valore del colore nella scala di grigio., ecc.. In questo caso ci possono essere più di n_P *parole-codice* dato che $n_W = 1$. Di conseguenza il numero massimo di colonne da decodificare sono $N_C = n_P$.

- *Codici time-multiplexing*: una sequenza di T trame viene proiettata e misurata in T istanti diversi. Le *parole-codice* associate ad ogni punto p_A è la sequenza di T diversi valori del pixel p_A. in questo caso ci possono essere più di n_P^T *parole-codice* ed il massimo numero di colonne da decodificare è $N_C = n_P^T$.

- *Codici spatial-multiplexing*: la *parola-codice* associata a ciascun pixel p_A è la distribuzione spaziale della trama in una finestra di n_W pixel centrata in

p_A. in questo caso ci possono essere oltre $n_P^{n_W}$ *parole-codice*. Per esempio, se una finestra ha 9 righe e 9 colonne, al quale corrisponde un codice comune, $n_W = 81$. E' importante notare come in questo caso i pixel vicini influenzano parte del loro codice, generando interdipendenza tra le codifiche.

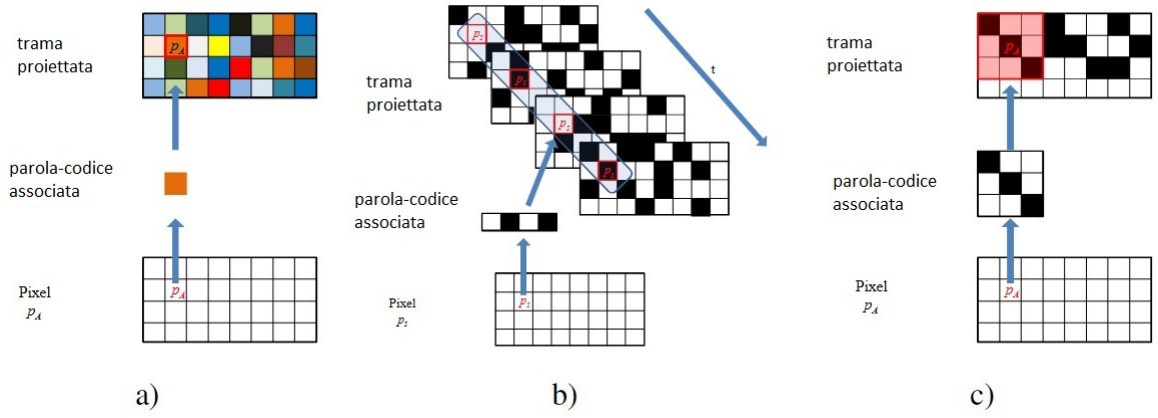


Figura 1.17: Strategie di codifica: a) codifica diretta; b) codifica time-multiplexig; c) codifica spatial-multiplexing

Ognuna delle strategie di codifica presentate hanno vantaggi e svantaggi. In particolare metodi di codifica diretta sono più semplici da implementare e più indicati per scene di movimento per il fatto che richiedono solamente la proiezione di una trama singola. Il loro svantaggio sta nella sensibilità alle distorsioni di colore o della scala di grigi ed alla maggior influenza della illuminazione esterna e delle non idealità delle telecamere.
I metodi *time-multiplexing* permettono di utilizzare un insieme ristretto di trame (e quindi di codici binari) al fine di aver comunque differenti *parole-codice* per ogni pixel. Il loro punto di forza sta nella robustezza alle distorsioni dei colori o scale di grigi, riflettività ed illuminazioni esterne. Lo svantaggio sta nella dovuta proiezione di T trame in T istanti differenti, il che rallenta la fase di acquisizione ed influenza l'utilizzo in scene movimentate.

Infine le strategie $spatial - multiplexing$ sono le più interessanti in quanto possono essere utilizzate anche in scene di movimento. Come i metodi di codifica

diretta, essi richiedono solamente la proiezione di una trama per decodificare l'immagine. Questi metodi uniscono i vantaggi delle altre due tecniche citate sopra. Le problematiche si riscontrano in presenza di occlusioni del fascio proiettato ed in caso di distorsioni della prospettiva. La scelta di un'adeguata finestra di dimensione n_W risulta quindi essere importante. Più piccola viene scelta la finestra e più robusto sarà il codice rispetto la distorsione della prospettiva in quanto meno pixel sono interessati ad indicare la stessa disparità; più grande viene scelta la finestra più si avrà robustezza alle non idealità dovute a telecamera e proiettore.

1.5.3 Approfondimenti riguardo il $Kinect^{TM}$

Questo paragrafo ha il compito di esplicitare delle analisi eseguite sul $Kinect^{TM}$ che potrebbero essere non del tutto esaurienti in quanto i suoi algoritmi sono del tutto sconosciuti, ma qualche caratteristica riguardo il linguaggio di decodifica può essere dedotto.

Incorrelazione della trama proiettata

Il sistema di rilevazione di profondità del $Kinect^{TM}$ utilizza una codifica *spatial-multiplexing* che permette di catturare immagini alla frequenza di 30 *fps*. La risoluzione spaziale della mappa di profondità in uscita è di 640x480 pixel, ma il sensore C ed il proiettore A hanno risoluzioni ancora sconosciute.

La trama proiettata (1.18)è caratterizzata dalla incorrelazione degli elementi lungo ogni riga. Questo significa che la covarianza tra le finestre utilizzate per il *spatial-multiplexing* centrate in un determinato punto p_A^i con coordinate $p_A^i = [u_A^i, v_A^i]^T$ e la trama proiettata nella finestra centrata in p_A^j con coordinate $p_A^j = [u_A^j, v_A^j]^T$, assumendo $v_A^i = v_A^j$ è 0 se $i \neq j$ ed è 1 se $i = j$.
Alcune analisi dimostrano che le dimensioni utilizzate pere la finestra sono 7 x 7, mentre altri studi suggeriscono una dimensione di 9 x 9 pixel (Vedi [3] e [4]).

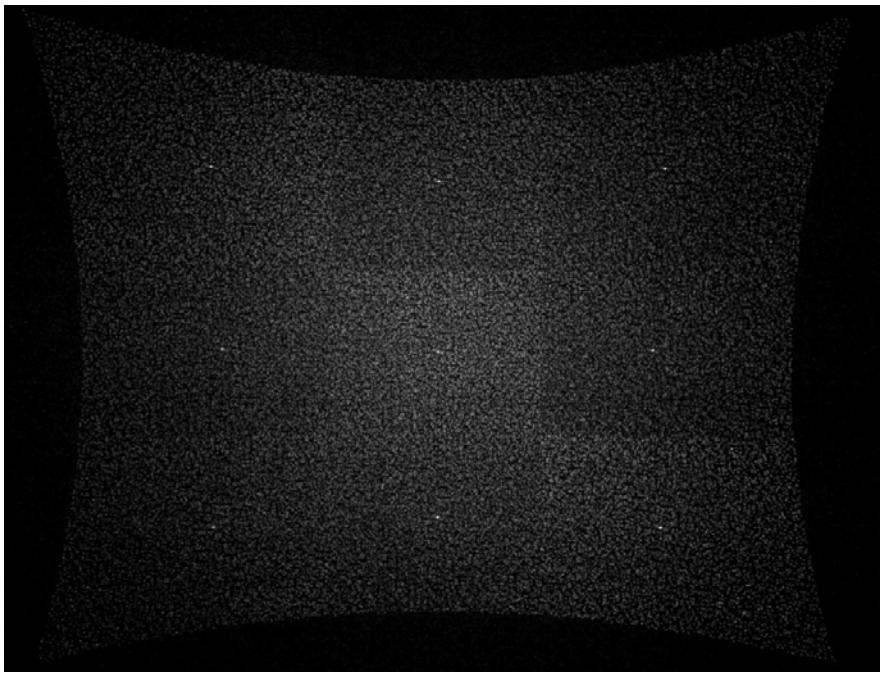


Figura 1.18: acquisizione ad alta risoluzione della trama proiettata. Il $Kinect^{TM}$ è stato posto di fronte ad una superficie piatta in condizioni di buio, l'acquisizione è stata eseguita da una telecamera standard.

Le trame misurate dal $Kinect^{TM}$ attraverso la telecamera IR sono affette, come aspettato, dalle distorsioni dovute alle non-idealità della telecamera e del proiettore. In questo contesto è stata eseguita una prova immettendo il $Kinect^{TM}$ di fronte ad una superficie piana al fine di rilevare la covarianza tra $p_A^i = [19, 5]^T$ e $p_A^j = [u, 5]^T$ con $u \in [1, 200]$. Si è ipotizzato una finestra 9 x 9.

Come ci si aspettava, tenendo conto che la trama acquisita è una versione distorta di quella proiettata, il grafico accentua un picco alla coordinata $u = 19$. Perciò, in questo caso, pe run pixel p_C^j di I_K e per tutti i pixel p_a^i della stessa riga della trama proiettata, la covarianza presenta solamente un picco in corrispondenza della coppia attuale di punti coniugati. In molte situazioni reali, i punti coniugati vengono scelti dal massimo valore di covarianza ottenuto nella stessa riga in quanto al fatto che, causa disturbi e non-idealità, non è sempre possibile ottenere covarianza unitaria. Il massimo della covarianza non è ben definito quando non ci sono punti coniugati causa l'occlusione o quando l'impatto di una

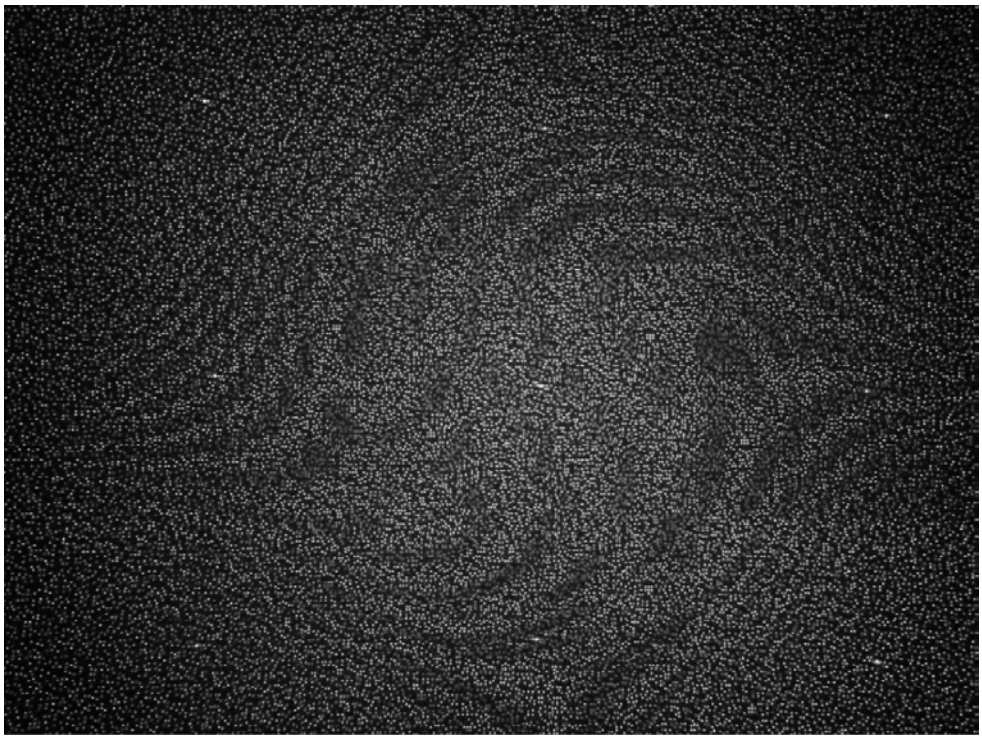


Figura 1.19: acquisizione della trama proiettata da parte della telecamera IR del $Kinect^{TM}$. l'acquisizione è eseguita al buio di fronte ad una superficie piatta.

sull'altra è troppo alta da rendere le *parole-codice* acquisite distorte (e quindi con alti valori di auto-covarianze):

Immagine di riferimento e problema della corrispondenza

Una comparazione tra le immagini I_K acquisite da C e la trama proiettata da A renderebbe visivi gli effetti dovuti alle non-idealità di ambedue i sistemi. Questo problema può essere ovviato con una procedura di calibrazione.
La procedura in questione richiede una acquisizione off-line in assenza di illuminazione esterna e di una superficie ad alta riflettività orientata ortogonalmente all'asse ottico della telecamera C e posizionata ad una determinata distanza. In questo caso la superficie acquisita è caratterizzata da una costante già conosciuta z_{REF} a da una costante di disparità d_{REF}. L'immagine acquisita è chiamata *immagine di riferimento.*
In qualsiasi acquisizione successiva è possibile associare ogni punto p_{REF} di coordinate $p_{REF} = [u_{REF}, v_{REF}]^T$ dell'immagine di riferimento con un punto p del-

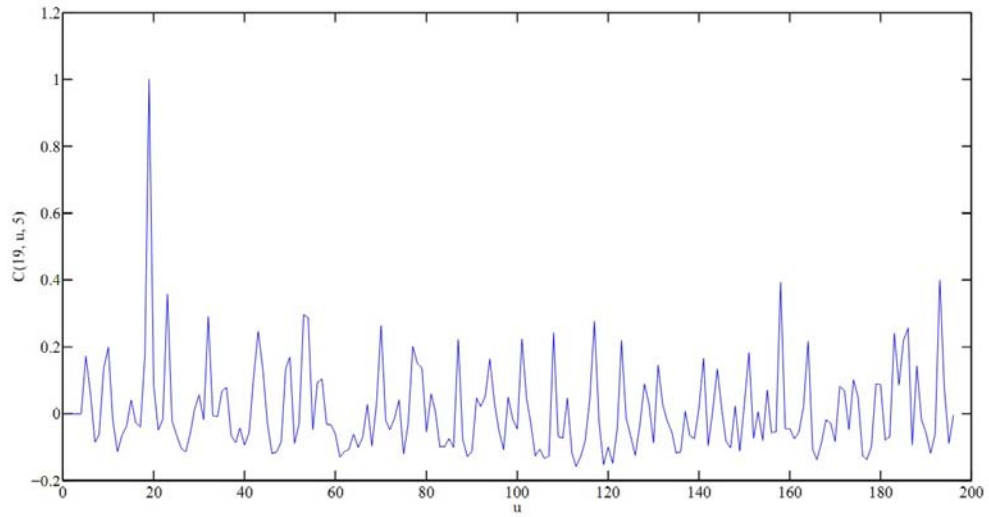


Figura 1.20: Covarianza della trama acquisita dal $Kinect^{TM}$ per il punto $p_A^i = [19, 5]^T$ e $p_A^j = [u, 5]^T$ con $u \in [1, 200]$.

l'immagine acquisita ed esprimere le sue coordinate rispetto al p_{REF} in questo modo: $P = [u, v]^T = [u_{REF} + d_{REL}, v_{REF}]$. In questo modo il valore di disparità d_A di ogni punto scena può essere calcolato aggiungendo d_{REF} alla disparità d_{REL} direttamente trovata dall'immagine acquisita.

$$d_A = d_{REF} + d_{REL}$$

L'uso di una immagine riferimento permette di ovviare ad alcune difficoltà indotte delle varie trasformazioni che distorgono la trama acquisita, sempre però, rispettando le non-idealità della proiezione.

In altre parole, il paragone dell'immagine I_K di una generica scena con l'immagine di riferimento è un metodo implicito di ovviare alle non-idealità a distorsioni dovute a C e A.
La stima della *disparità* è una procedura del tutto simile a quella già vista nel caso stereoscopico. In questo caso, stimare il pixel coniugato dal massimo della covarianza è solamente un algoritmo *locale*, nel senso che non interferisce al resto della computazione in quanto analizza solo un gruppo di coppie possibili e de-

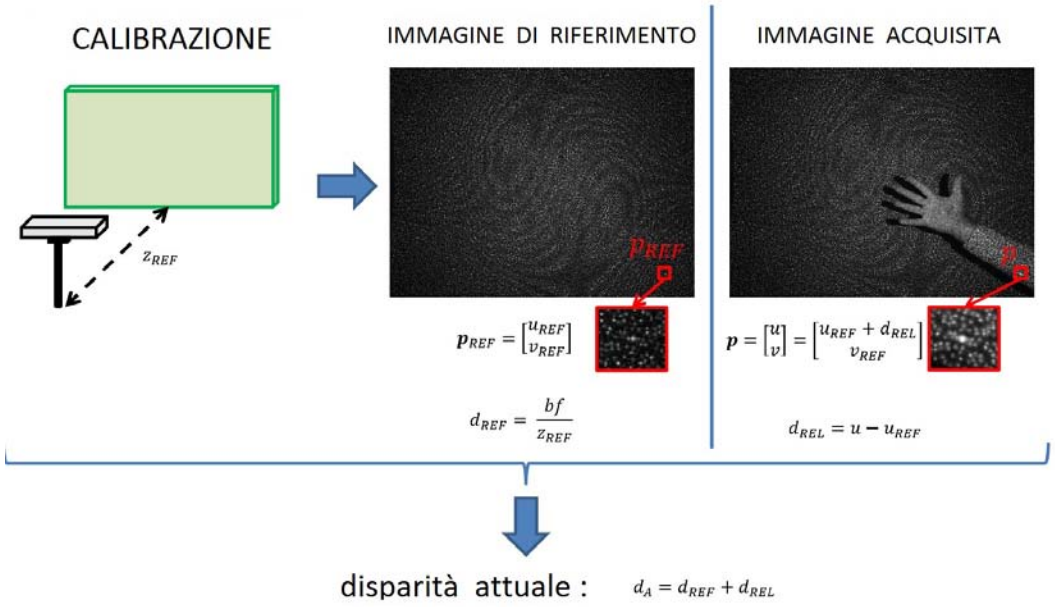


Figura 1.21: Una tecnica di calibrazione del $Kinect^{TM}$: al centro l'immagine di riferimento posta in perpendicolarità del dispositivo ed il calcolo did_{REF} , a destra l'immagine di una scena acquisita ed il calcolo di d_{REL}.

termina quale da vita alla miglior similarità. Come già visto ci sono una marea di altre tecniche per analizzare non solo alcuni punti appartenenti alla linea ma molti altri (seguendo un schema di ottimizzazione globale). E' probabile che anche il $Kinect^{TM}$ adotti algoritmi di questo genere.

Miglioramento attraverso i *sub-pixel*

Una volta che tutti i valori di *disparità* dei pixel di I_K sono stati calcolati è possibile qualche affinamento. In particolare, tutti i valori di disparità trovati attraverso massimizzazione della covarianza o altre tecniche sono assunti implicitamente come *integer*. Dalle metodologie utilizzate nella stereovisione è ben noto che ciò limita la risoluzione della profondità e questo ultimo valore può essere ottimizzato con le tecniche di raffinamento *sub-pixel* (che però aumentano notevolmente la complessità compiutazionale). In accordo con il testo [4] anche il $Kinect^{TM}$ utilizza tecniche di miglioramento di questo tipo.

1.5.4 Dati tecnici

In accordo con [2] riporto alcuni chiarimenti riguardo l'accuratezza del dispositivo. E' stato scritto uno script in *Matlab* al fine di rilevare la risoluzione spaziale e la risoluzione nella distanza sapendo avendo a disposizione misurazioni reali degli stessi.
Una prima prova è stata eseguita sapendo le dimensioni reali di una zona inquadrata e contando i pixel che la formavano. da qui il $Kinect^{TM}$ è stato spostato di dieci in dieci centimetri.

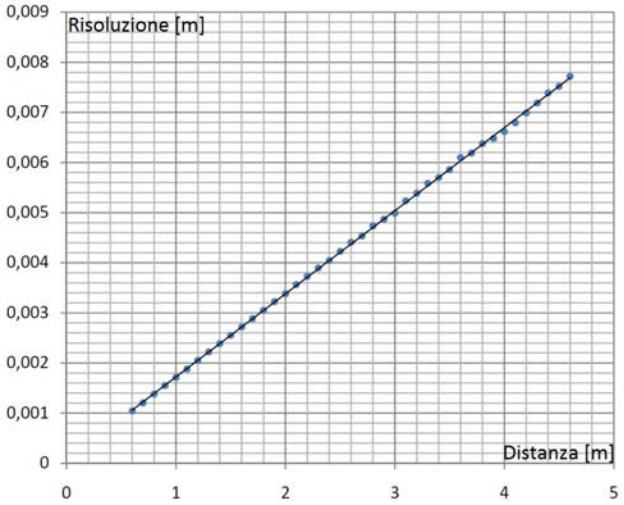


Figura 1.22: Risoluzione spaziale di x e y in riferimento alla distanza in metri.

Per quanto riguarda la risoluzione di profondità, è noto che il $Kinect^{TM}$ fornisce dei dati già espressi in unità metriche ad 11 bit. Da una prova pratica questo è ciò che ne deriva:

1.5.5 Problemi pratici nell'acquisizione

Come già spiegato nel paragrafo precedente ci sono molte fonti di errori che negano alcune corrispondenze ed altri che si traducono in valori errati di profondità. Alcune fonti di questi errori sono la telecamera ed il proiettore, alcuni dall'algoritmo di stima della corrispondenza ed altri dovuti alla geometria della scena inquadrata. Da un'analisi sperimentale dei dati forniti dalla telecamera IR del

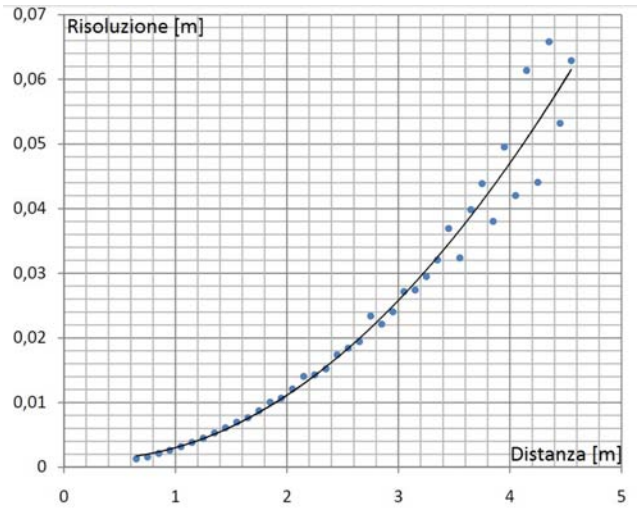


Figura 1.23: Risoluzione di profondità in relazione con la distanza.

$Kinect^{TM}$, gli errori più rilevanti sono dovuti a:

- *Bassa riflettività e illuminazioni di fondo.* Nel caso di bassa illuminazione e illuminazioni di fondo la telecamera è incapace di acquisire nessuna informazione riguardo la parte di trama riflessa ed il problema della corrispondenza non produce alcun risultato. Quindi in questi casi ci sono perdite di informazioni di profondità.

- *Superfici eccessivamente inclinate.* in questo caso la distorsione prospettica è talmente forte che nella finestra determinata per lo *spatial-multiplexing* ci sono troppi pixel caratterizzati tra disparità differenti. Anche in questo caso l'algoritmo di corrispondenza non dà risultati e viene a mancare l'informazione sulla profondità.

- *Occlusioni e discontinuità.* Vicino alle discontinuità nella profondità la finestra usata nello *spatial-multiplexing* può includere pixel associati a disparità differenti, con conseguente generazione di errori. Ad aggiungere a tal fatto ci sono occlusioni dovute a particolari superfici e forme che impossibilitano i punti proiettati di raggiungere la zona visibile dalla telecamera IR.

Il $Kinect^{TM}$ adotta una propria Euristica e calcola di propria iniziativa i valori mancanti interpolando i valori dei punti vicini. Questa assegnazioni portano a degli disallineamenti tra le discontinuità dell'immagine reale e dell'immagine ricreata (fino a 10 pixel).

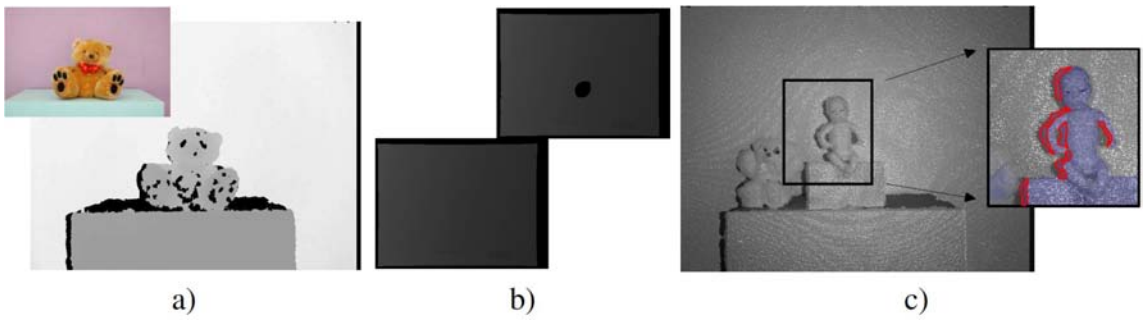


Figura 1.24: Differenti errori nella stima della profondità: a) basso fattore di riflessione della superficie dell'orsacchiotto e superficie del tavolo troppo inclinata; b) superficie acquisita con e senza retroilluminazione; c) discontinuità nella profondità sono ben visibili nel lato sinistro della bambola.

La risoluzione spaziale del $Kinect^{TM}$, anche se non è ben nota la risoluzione del sensore interno, deriva dal fatto che le matrici I_K,$\hat{D}_K$ e $\hat{Z}_K$ hanno una dimensione di 640 x 480.
Vale la pena ricordare un altro problema legato alla triangolazione. La risoluzione della profondità stimata decresce con la radice della distanza. Pertanto la qualità delle misurazioni della profondità ottenute dalla triangolazione matriciale sono peggiori per le scene lontane rispetto alle scene vicine.
Infine è bene riportare una particolare caratteristica del rilevamento della profondità del $Kinect^{TM}$: durante gli esperimenti è stato provato che ad ogni movimentazione il dispositivo impiega almeno 30 secondi per fornire senza varianza temporale il livello di profondità. Infatti, preso un pixel a caso nell'immagine, si nota che il suo valore *depth* varia da un'acquisizione ad un'altra, stabilizzandosi dopo un certo periodo di tempo. La figura 1.5.5 aiuta a capir meglio la cosa.

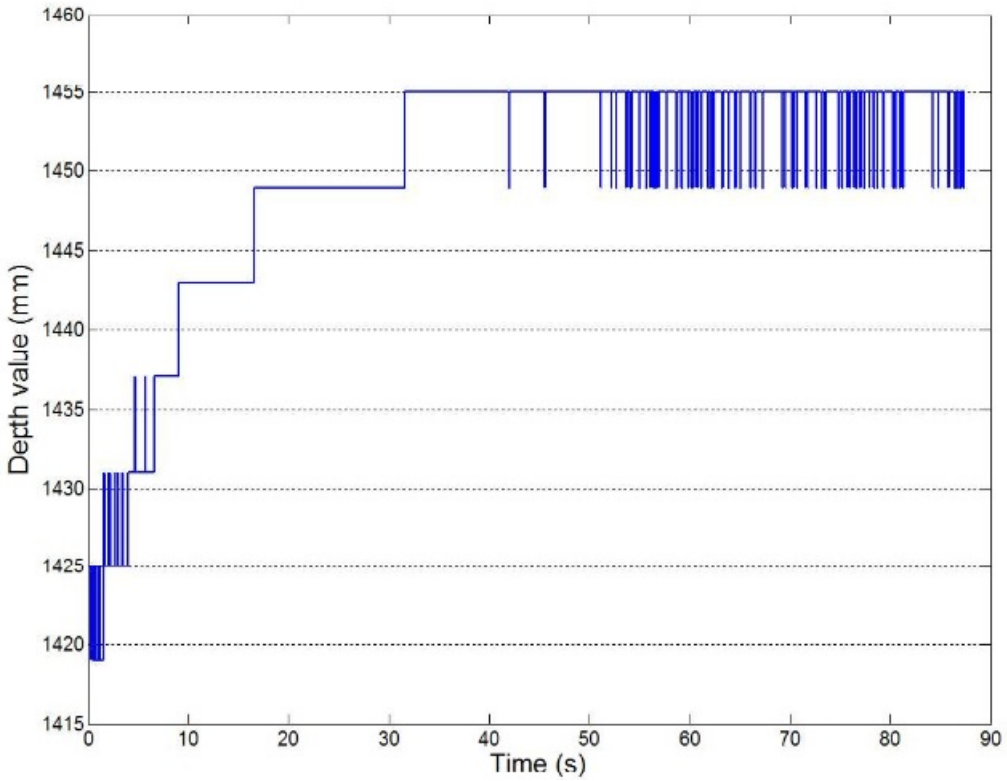


Figura 1.25: L'immagine raffigura il valore dei pixel in funzione del tempo trascorso dopo una movimentazione [2].

Capitolo 2

Adept Quattro^{TM}

2.1 Introduzione

I robot industriale presenti in commercio si distinguono per gradi di libertà, dimensioni e disposizione degli assi.
Una distinzione può essere fatta tenendo conto delle configurazioni:

- Robot seriali, costituiti da una catena di giunti disposi uno in serie all'altro:
  - Robot SCARA; i primi tre giunti rotazionali sono disposti verticalmente. All' *end-effector* viene permessa la rotazione e la traslazione in verticale;
  - Robot PUMA; essi richiamano una arto superiore umano (detti anche antropomorfi). La presenza di sei assi permettono il loro utilizzo per l'assemblaggio;
  - Robot CARTESIANI; costituiti principalmente da tre *giunti prismatici*.
- Robot paralleli:
  - Robot Delta; costituiti da 3 coppie di bracci di egual lunghezza. La movimentazione delle tre coppie permette di muovere l'*end-effector* lungo le coordinate x, y, z e praticarne l'orientazione sono nell'asse z;

– Robot Quattro; stesso principio del Robot Delta, ma utilizza 4 coppie di bracci. Questa soluzione è stata intrapresa dall'americana *Adept*.

2.2 Adept Quattro^{TM} S650

Adept Quattro S650 è un robot parallelo a 4 assi. I 4 motori piazzati alla base controllano 4 assi che permettono di generare una movimentazione in x, y e z ed a garantire la rotazione nell'asse z.
Nella base del robot viene integrata l'elettronica di potenza *Adept SmartServo*, mentre è richiesta la connessione del robot all'unita *Adept SmartController*.

Figura 2.1: Adept Quettro S650: foto

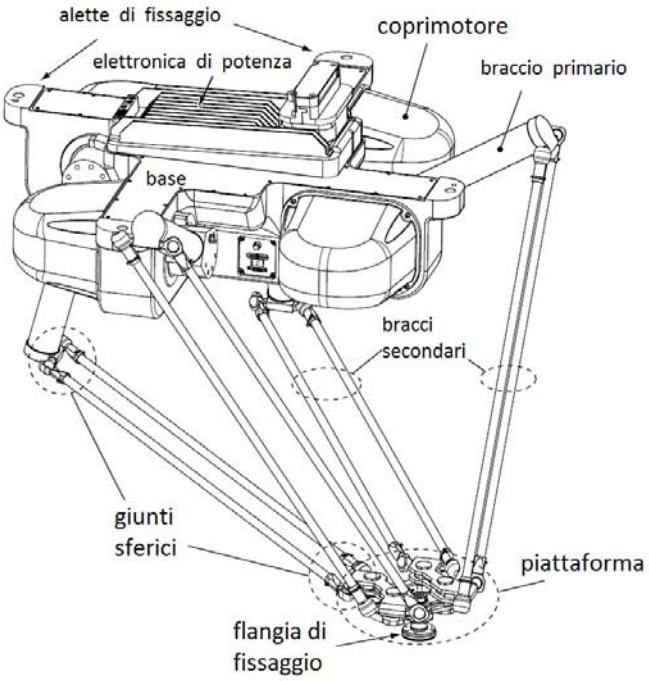


Figura 2.2: Adept Quettro S650: gli organi principali

Le caratteristiche tecniche del robot ne indicano l'impiego nella palettizzazione. Infatti il robot consente movimentazioni rapide (velocità fino a 10 *m/s*) e ripetibili ma non accurate come nei robot Scara o Viper.

2.2.1 Base del robot

La base del robot Adept Quattro S650 è costruita in alluminio e permette di alloggiare i quattro motori oltre che all'elettronica di potenza AIB (Amplificatori-in-base). Essa provvede ad una buona conduzione termica al fine di raffreddare motori e l'unità di amplificazione. In essa viene alloggiato un pannello Led di controllo per segnalare lo stato del robot e la presenza di guasti e/o malfunzionamenti.

2.2.2 Adept AIB

Con la sigla AIB si intende un'unita di amplificatori che permettono, oltre che a convertire l'alimentazione alternata a 230V in tensione continua, di trasformare il segnale digitale proveniente dal *controller* in un segnale elettrico di potenza atto a generare la rotazione dei motori.

Figura 2.3: sistema AIB installato in Adept Quattro

L'assemblamento nella base del robot permette di avvicinare in maniera significativa la tratta di trasmissione di potenza e quindi limitare le perdite. Questa

unità integra inoltre dei controlli a retroazione. Ecco le principali caratteristiche:

- segnali digitali nella scheda: 12 ingressi, 8 uscite;
- basso rumore nei segnali: per l'utilizzo con attrezzature sensibili al rumore;
- no ventilazione;
- controllo di temperatura e corrente in tutti i motori;
- conversione in corrente continua.

2.2.3 Braccia del robot e piattaforma

Noto il fatto che i motoriduttori trasmettono la coppia ai bracci principali, il moto viene poi trasferito ai bracci secondari, che sono connessi con giunti sferici ad alta precisione. Questi otto braccia sono costruiti in fibra di carbonio. La piattaforma è collegata in maniera identica agli otto bracci.

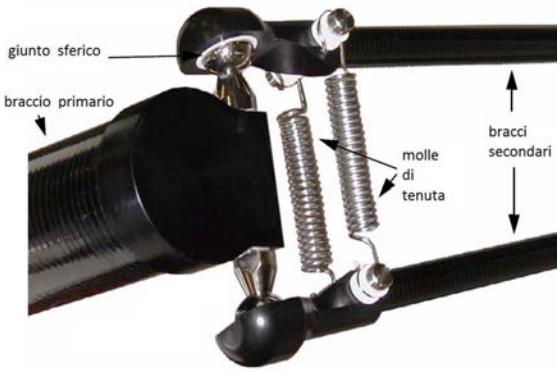


Figura 2.4: giunti sferici tra bracci

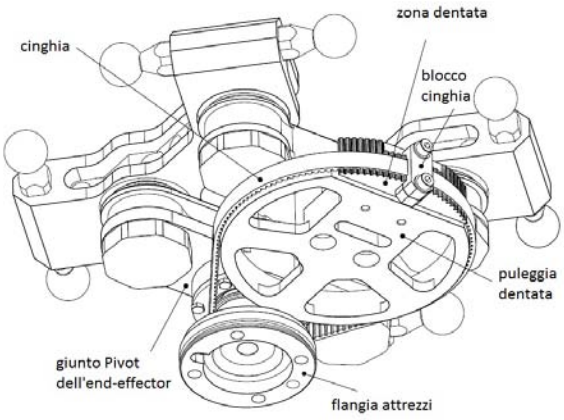


Figura 2.5: giunti sferici tra bracci secondari e piattaforma

Le otto braccia di carbonio sono tenute sotto pressione ai giunti sferici tramite delle molle che consentono, nel caso d'urto dell'*end-effector* di distaccarsi momentaneamente preservando la meccanica del robot.
La particolare configurazione della piattaforma consente una rotazione di $\pm 180°$

con una precisione di 0.4°, mentre la precisione nelle tre lunghezze x, y, z è dichiarata $0.1mm$.

2.2.4 Adept SmartController CX

Lo Smartcontroller costituisce l'organo di comunicazione tra tutte le periferiche ed il robot. Per la comunicazione *real-time* e per la programmazione sfrutta la connessione Ethernet con il PC.

Figura 2.6: Adept SmartController CX

Oltre a ciò è possibile connettere il *pendant*, il modulo di emergenza, controllo del nastro trasportatore ecc.

2.3 Cella di lavoro

La cella di lavoro del robot *Adept Quattro* è costruita in profilati di alluminio ed è munita di nastro trasportatore. Le dimensioni consentono quasi il totale movimento del robot nello spazio (a parte una zona occupata dal nastro). La cella consiste in una struttura portante il robot, per cui la sua rigidezza elevata ha permesso una regolazione in orizzontalità in concomitanza col robot.
La cella è stata ridisegnata con il software SolidWorks a partire da ogni singolo profilato e guarnizione rispettando le misure reali. Il ciò permetterà in seguito di eseguire un confronto con il file *.stl o *.ply generato dalla ricostruzione tramite telecamera 3D.

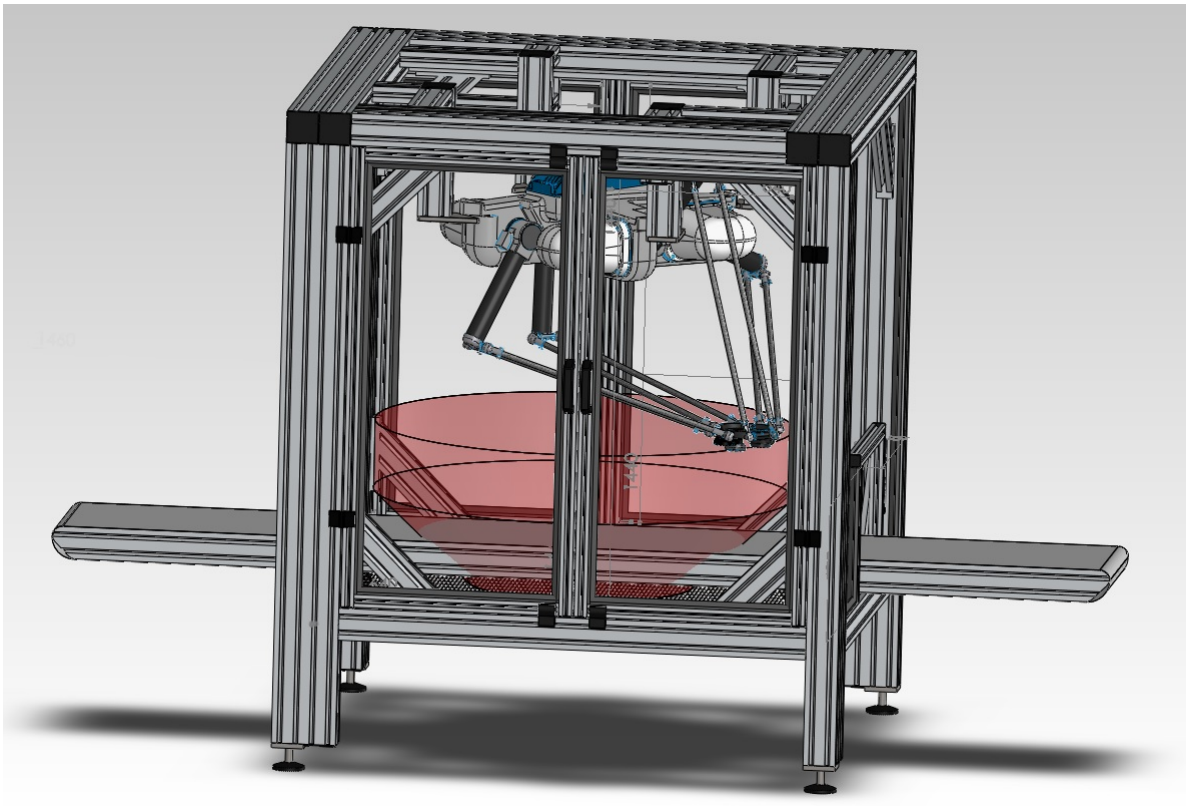


Figura 2.7: cella robot con Adept Quattro s650 e volume di lavoro in evidenza.

Capitolo 3

Bin Picking

3.1 Introduzione

In questo testo si vuole dare un'introduzione a ciò che è oggi l'applicazione più diffusa della visione tridimensionale: il *Bin Picking* . Con questo termine si intende la presa di un oggetto disposto in maniera casuale in un contenitore nel quale sono presenti altri oggetti disposti alla rinfusa.
Il problema presenta difficoltà via via crescenti a seconda le tipologie di oggetti presenti siano una o più ed a seconda che la disposizione sia ordinata o random. In un primo momento si introdurrà il problema della localizzazione 2D per poi ampliare la trattazione nella localizzazione 3D. Si farà riferimento ai sistemi di visione adottate ed alle tecniche computazionali.

3.2 Localizzazione 2D

Nel caso di oggetti di costruzione piana (es. laminati) od oggetti il cui posizionamento è tale che sia sempre la stessa faccia ad appoggiare sul piano è possibile effettuare la localizzazione semplicemente a partire da un'immagine 2D.

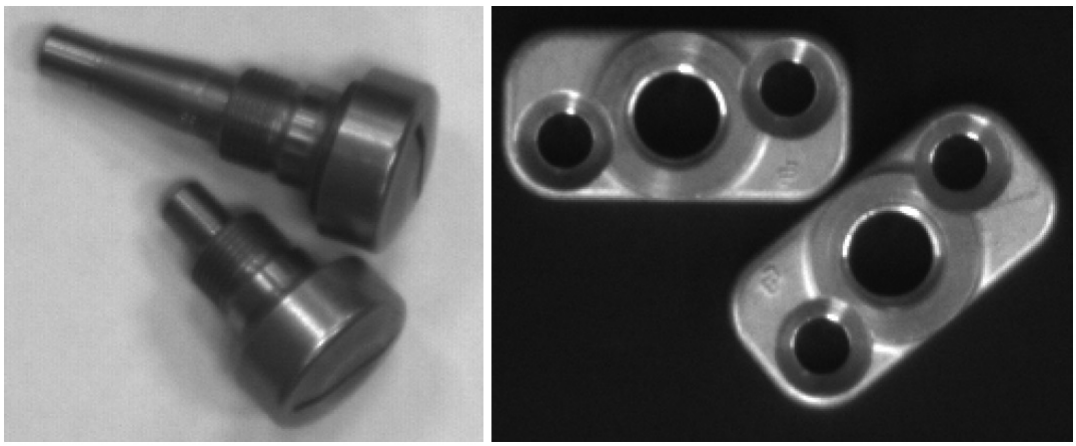


Figura 3.1: oggetti in fase di presa con il sistema 2D

Con questi metodi si assume che la coordinata z per la presa dell'oggetto sia sempre la stessa, per ottenere quindi un corretto posizionamento si sfruttano talvolta dei vibratori industriali garantendo che gli oggetti non siano sormontati. Viene lasciata all'immagine della telecamera (posta con l'asse dell'ottica verticale al piano) il compito di rilevare la posizione (x,y) dell'oggetto. Per far ciò si cercano ovviamente le geometrie dell'oggetto attraverso algoritmi di filtraggio dell'immagine; una volta individuata la geometria è possibile confrontarla con le geometrie in memoria. E' cosi possibile effettuare la presa nella modalità ottimale per l'oggetto considerato.

3.3 Localizzazione 3D

Le applicazioni di robotica più complesse, come ad esempio quelle in cui la posizione dell'oggetto da localizzare non è determinata, ma può variare in un range continuo, il sistema deve essere in grado di localizzare e fornire al robot la posa completa 3D (posizione e orientamento sui tre assi coordinati) dell'oggetto localizzato. Questo è il caso di oggetti sormontati tra di loro o messi alla rinfusa in un contenitore.

Figura 3.2: oggetti in fase di presa con il sistema 3D

Per ottenere la posa 3D di un oggetto esistono diversi approcci:

3.3.1 Sistemi a singola telecamera standard

E' noto che al fine di ottenere informazioni 3D complete, è necessario disporre di un sistema stereoscopico (immagini dell'oggetto di interesse provenienti da due punti di vista differenti). Ciononostante, fornendo le informazioni mancanti per altra via, è possibile ottenere la terza dimensione anche utilizzando sistemi a singola telecamera.
Se la superficie dell'oggetto di interesse ha una trama ben visibile, mediante la tecnica della localizzazione delle deformazioni è già possibile ottenere la posa

completa 3D semplicemente localizzando una immagine prememorizzata. Se le superfici dell'oggetto non si prestano ad una localizzazione affidabile, si può ottenere la posa 3D con una potente tecnica di localizzazione che, a partire da una singola immagine e dal disegno CAD dell'oggetto, è in grado di fornire in brevissimo tempo le 6 coordinate di posizione ed orientamento. Inoltre, risulta ancora più semplice identificare la posa completa 3D, se l'oggetto ricercato mostra una primitiva facilmente identificabile (un cerchio o un rettangolo): disponendo della sola dimensione (raggio del cerchio o lati del rettangolo) è possibile localizzare posizione ed orientamento 3D dell'oggetto in tempi dell'ordine di poche decine di millisecondi.

3.3.2 Sistemi stereo

I sistemi stereo (che utilizzano due telecamere standard) permettono di ottenere una ricostruzione 3D della scena ripresa, ovvero di determinare le coordinate 3D di ogni punto dell'immagine. In questo modo diventa possibile eseguire tutte le misure necessarie direttamente in 3D per ottenere la posa dell'oggetto. In generale questa tecnica consente di realizzare ispezioni 3D generiche. Combinando insieme diverse tecniche è possibile risolvere problemi anche molto complessi. Presentano l'inconveniente che in molte applicazioni il contenitore in cui sono presenti gli oggetti è scarsamente illuminato ed il problema della corrispondenza diventa quindi aleatorio.

3.3.3 Sistemi a singola telecamera 3D

Una tecnica di visione 3D utilizzata nel *BinPicking* è la triangolazione laser. Nello stesso ambito vengono utilizzate anche le telecamere ToF descritte al capitolo [2]: tramite queste telecamere è possibile conoscere direttamente la posizione dei punti dell'oggetto in coordinate tridimensionali. Il vantaggio di questi sistemi di visione consiste nel disporre di illuminazione propria e quindi di essere affidati anche in luoghi di bassa luminosità: si toglie quindi la dipendenza dal colore. Ven-

gono privilegiati i sistemi laser nel caso si necessiti di un'alta risoluzione, mentre in applicazioni in cui si richiede un'acquisizione rapida si utilizzano telecamere ToF.

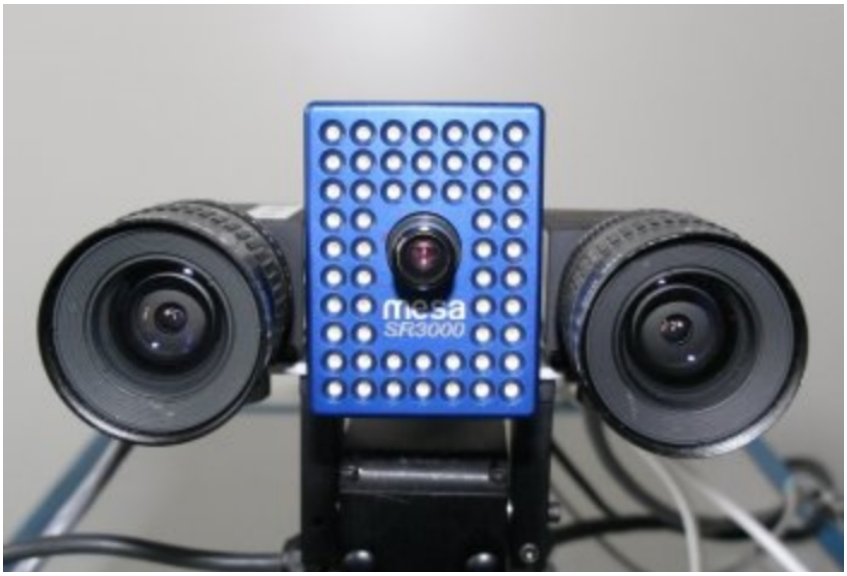


Figura 3.3: sistema ibrido: Mesa SR3000 e due telecamere RGB standard.

3.3.4 Sistemi ibridi

In alcune applicazioni si integra un sistema di visione 2D e 3D al fine di rendere più efficiente l'operazione di rilevamento della posizione dell'oggetto. La visione 2D permette di selezionare facilmente l'oggetto e quindi, attraverso la sua geometria, rilevare la posizione in cui si effettuerà la presa. In un secondo momento la telecamera 3D, solitamente posta all'*end-effector* del robot, risulterà efficace per rilevare l'orientazione della presa.

3.4 Elaborazione

Il cuore del *binPicking* sta nell'elaborazione dei dati forniti dalle immagini e dai modelli CAD pre-esistenti. Infatti senza una base di confronto l'algoritmo non può riconoscere l'oggetto e quindi non è possibile la presa.
La prima fase dell'algoritmo consiste nell'acquisizione di una più immagini al fine di creare un modello 3D della zona del *picking*. In taluni casi infatti, l'utilizzo di telecamere stereo o telecamere fissate al robot permettono di aver più fotogrammi che inquadrano la scena da punti diversi. L'utilizzo di algoritmi ICP (Iterative Closest Point) permette di connettere due nuvole di punti in modo da formare un unico modello 3D. Il tutto si basa sulla minimizzazione della distanza tra i punti di due immagini differenti. In questo modo è possibile ottenere la matrice di rototraslazione per la seconda nuvola di punti e quindi ricreare il modello.

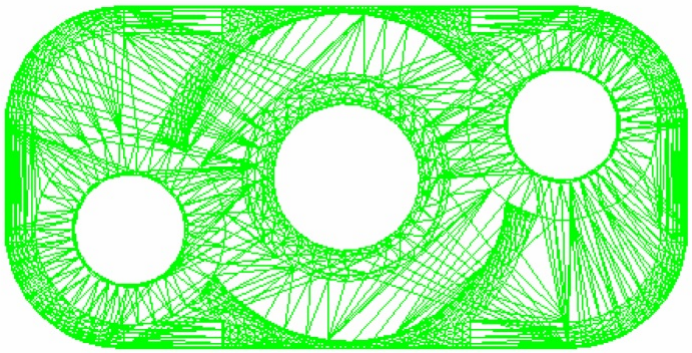


Figura 3.4: modello CAD di un pezzo da prendere

La seconda fase consiste nel trovare un modello matematico sensato nella nuvola di punti. La GHT (Generalized Hough Transform) è uno dei metodi più diffusi in quanto permette di trovare semplici modelli geometrici a partire dalla trasformata di Hough. Ciò è possibile dando importanza ai pixel che sembrano formare il profilo del pezzo considerato; ricreata una linea di congiunzione si può riproporre una probabile forma geometrica.
Ora sta ad un algoritmo di *matching* far sovrapporre modello CAD e modello ricreato in modo da individuare la matrice di rototraslazione che mi definirà il movimento per la presa.

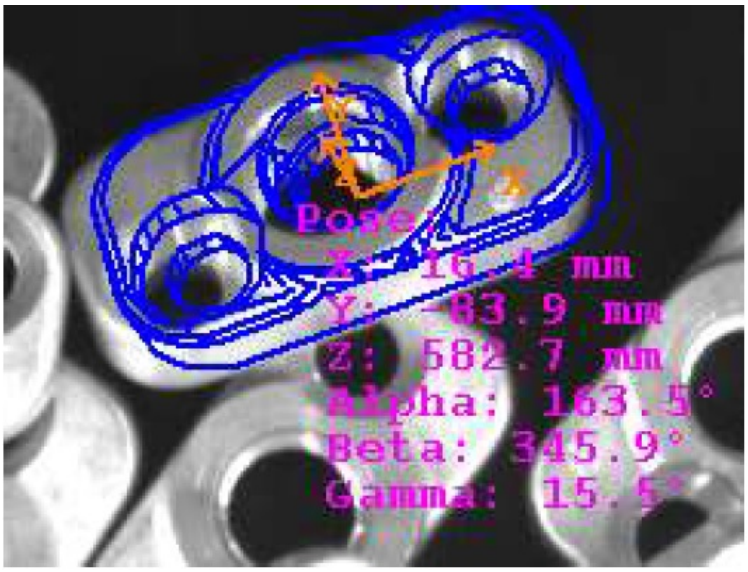


Figura 3.5: Matching tra il modello CAD e l'immagine

Un problema che i recenti algoritmi affrontano è la predizione della presa senza problematiche. In alcuni casi infatti non è possibile raggiungere l'oggetto poiché la posizione di presa non è effettuabile dal robot, in altri compaiono ostacoli fisici quali il contenitore o addirittura l'ingombro da parte di altri oggetti da prendere. E' compito di un algoritmo robusto determinare se la presa è possibile e quindi giungere al termine con il maggior numero possibile di pezzi prelevati.

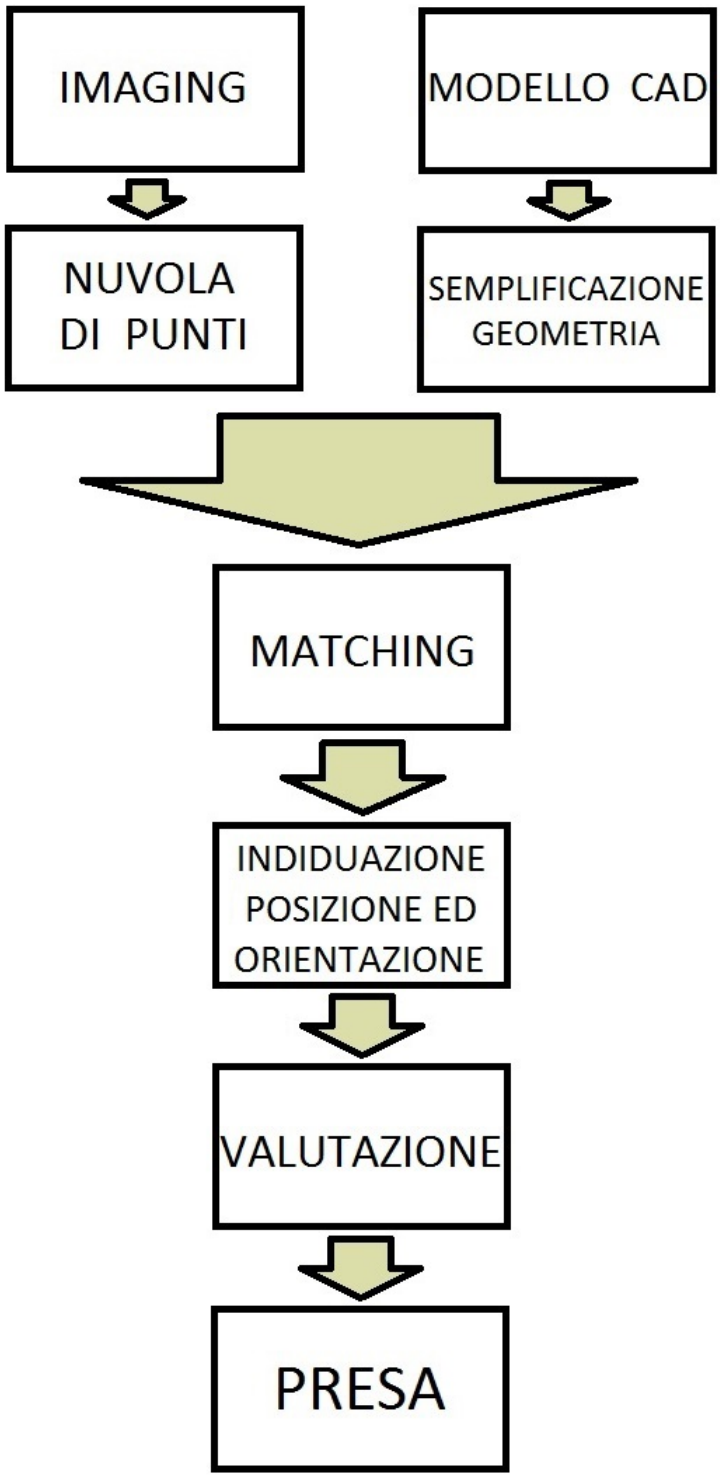


Figura 3.6: schema di massima degli algoritmi *Bin Picking*.

Capitolo 4

Ricostruzione grafica tramite Kinect e robot Adept Quattro s650

4.1 Introduzione

In questo capitolo viene trattata l'attività svolta presso i laboratori del DIMEG. La ricerca fatta si pone come base per la futura applicazione di una telecamera 3D all'*end-effector* del robot ai fini del *Bin-Picking*. E' stato impiegato il robot *AdeptQuattros*650 per muovere il $Microsoft\ Kinect^{TM}$ fissato alla flangia. Il robot permette 4 gradi di libertà, per cui sono possibili tutte le posizioni entro i limiti fisici e la sola rotazione nell'asse verticale z. Inoltre il dispositivo di *tilt* del $Kinect^{TM}$ permette un ulteriore grado di libertà nel movimento. Lo scopo sarà quindi di poter acquisire immagini di profondità della cella al fine di rilevare la presenza di oggetti e di crearne un modello importabile dai programmi di disegno 3D.

In una prima parte viene presentato il programma di perlustrazione del volume di lavoro del robot. In secondo luogo viene spiegato il metodo di acquisizione dei fotogrammi e di ricostruzione delle immagini di profondità con le problematiche incontrate. L'algoritmo di ricostruzione della scena si baserà su una nuvola di punti creata sapendo a priori la posizione del centro della telecamera e le infor-

mazioni di profondità.
In un secondo contesto si procederà a dare un'analisi qualitativa a ciò che è il sistema di acquisizione *Kinect-Robot*. Ciò è stato fatto confrontando il modello ottenuto con l'ausilio dell'ambiente di programmazione $Matlab^R$ con un modello generato da un sistema professionale di acquisizione (metodologia: triangolazione laser) avente un grado di accuratezza dell'ordine dei centesimi di millimetro.

4.2 Algoritmo di perlustrazione

La prima fase del lavoro consiste dunque nella realizzazione di un algoritmo per la movimentazione del robot. Grazie alla flessibilità del programma $Matlab^R$ è stato possibile interfacciare l'ambiente di programmazione *Adept Desktop* permettendo di operare quindi da un unica interfaccia. Inoltre è stata resa possibile la comunicazione col $Kinect^{TM}$ allo scopo di acquisire le immagini di profondità (Depth) e RGB solo nei momenti desiderati.

4.2.1 Codice Adept V+

Con l'ausilio di un file *Matlab Executable* è stato scritto il codice in *Adept DeskTop* per gestire la movimentazione robot negli istanti in cui *Matlab* invia un pacchetto contenente posizione ed orientazione. Essendo il robot progettato per i lavori di *pick and place*, quindi rapido nei movimenti, ne è stata limitata la velocità al 6 % . Ad ogni movimentazione eseguita un pacchetto fornirà al codice *Matlab* il via libera per richiedere al $Kinect^{TM}$ l'allineamento in orizzontale e la scansione dell'immagine.

4.2.2 GUI Scansione

Con il supporto di $Matlab^R$ è stata creata una GUI, ovvero un interfaccia grafica, per l'acquisizione delle immagini. Si è pensato di adattarla per qualsiasi tipologia di scansione, a partire dalla movimentazione manuale singolo punto alla movimentazione automatica di un insieme di punti. Le traiettorie nello spazio possono

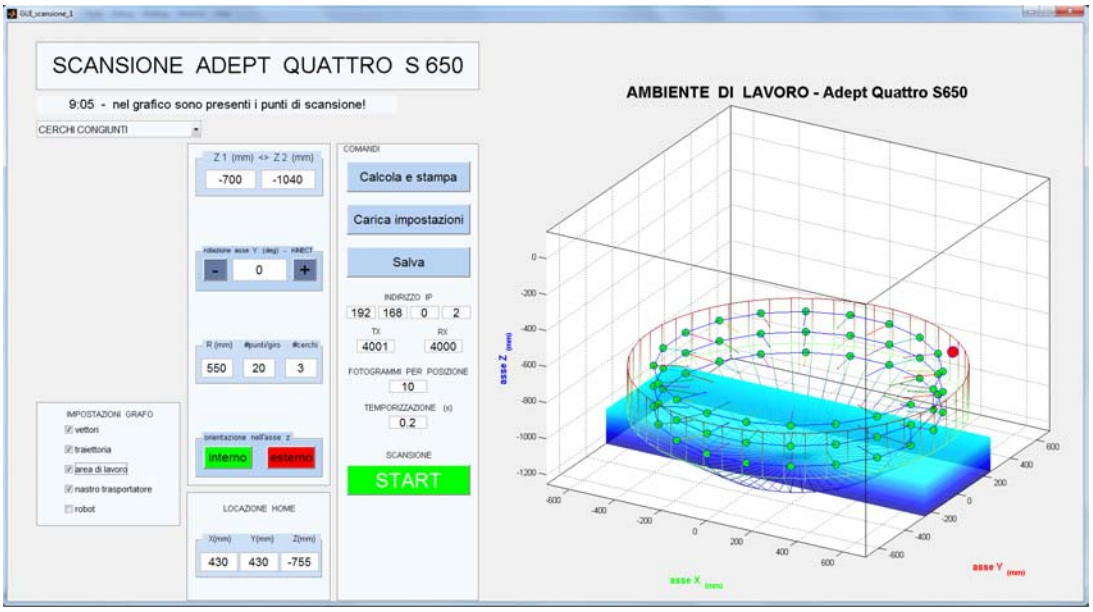


Figura 4.1: GUI di acquisizione

formare cerchi congiunti, sinusoidi o reticoli. L'orientazione dell'ultimo membro della catena cinematica viene stabilita in base all'utilizzo: fissa, rivolta verso il centro cella o verso l'esterno.

Come si nota in figura la GUI permette di scegliere il numero di acquisizioni *depth* per ogni posizione ed il tempo di attesa tra l'acquisizione di un fotogramma e l'altro.

Per la stabilità del programma sviluppato in ambiente *Adept Desktop* viene implementata una funzione di decisione della coerenza dei punti al fine di evitare che vadano fuori dallo spazio operativo del robot considerato: un eventuale punto non accettato genererebbe un errore e il blocco dell'esecuzione del codice.

Data la forma geometrica della zona di lavoro (vedi capitolo 2) vengono accettati solamente i punti che rispettano le seguenti regole matematiche:

$$\begin{cases} z_{MIN} < z < z_{MED} \\ R = \sqrt{x^2 + y^2} \\ R < R_{SUP} - \frac{(R_{SUP} - R_{INF})(z_{MED} - z)}{z_{MED} - z_{MIN}} \end{cases}$$

$\cup$

$$\begin{cases} z_{MED} < z < z_{MAX} \\ R = \sqrt{x^2 + y^2} \\ R < R_{SUP} \end{cases}$$

Una volta definito l'insieme di punti, essi vengono ordinati in modo da ottenere una minima movimentazione da un punto all'altro. Per esempio, eseguita una traiettoria a cerchio in senso orario, viene eseguita la successiva in senso antiorario evitando il passaggio nella zona centrale dell'area di lavoro. Nel caso del reticolo tridimensionale, esso è formato da più matrici di punti una sotto l'altra: il programma esegue un movimento a serpentina per completare il primo reticolo per passare al secondo con un sol movimento nell'asse z.

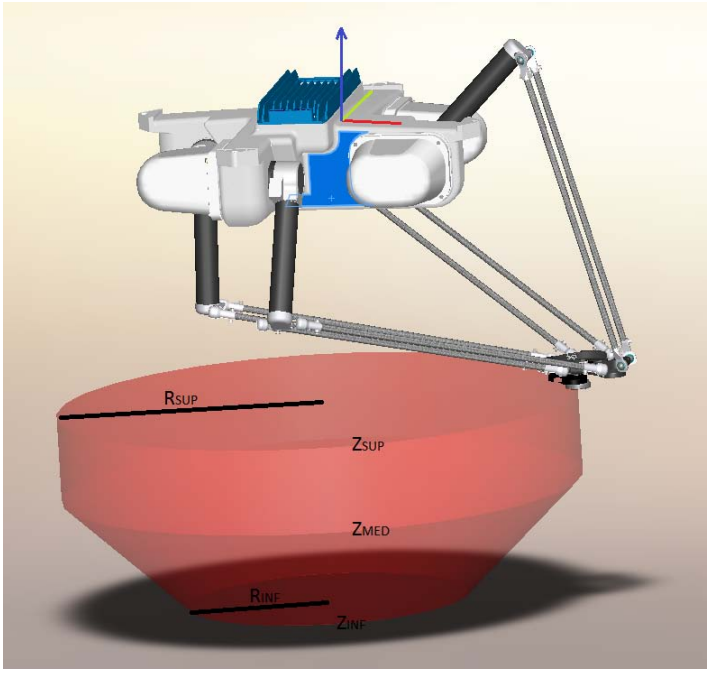


Figura 4.2: Modello Solidworks del robot Adept Quattro s650 e spazio di lavoro.

4.3 Calibrazione Kinect

Per la visione 3D è necessario calibrare la telecamera al fine di estrapolare tutte le informazioni metriche di un'immagine bidimensionale.
La tecnica utilizzata per calibrare la telecamera richiede di utilizzare almeno due

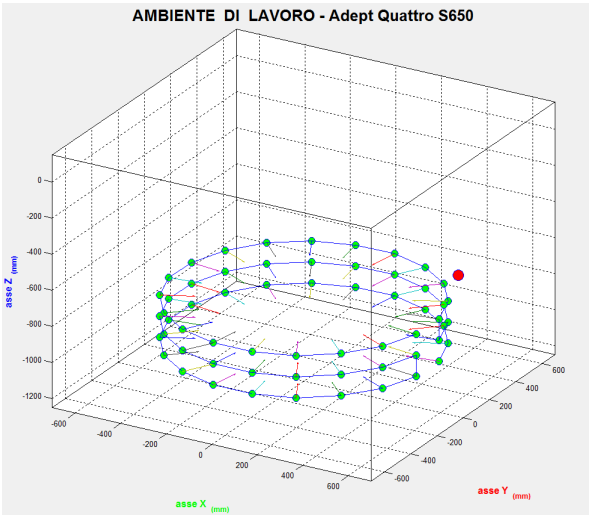


Figura 4.3: Movimento a cerchi congiunti con end-effector che punta all'interno

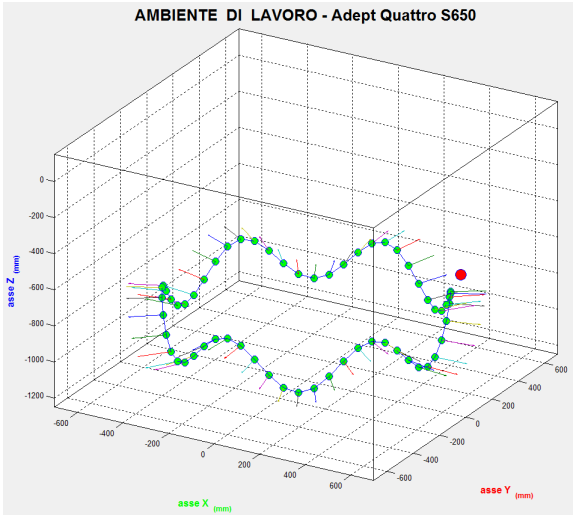


Figura 4.4: Movimento sinusoidale con end-effector che punta all'esterno

immagini di riferimento che inquadrano una trama appositamente disegnata senza aver conoscenza delle posizioni o dell'orientazione della stessa. La trama utilizzata consiste in una scacchiera in colorazione bianco e nero stampata su carta e posizionata in un piano rigido per evitare la deformazione. E' possibile muovere telecamera e trama senza problemi alla risoluzione. La procedura consiste in una soluzione in *forma-chiusa* seguita da un raffinamento dovuto al criterio di *massima-verosimiglianza*; l'approccio viene considerato una fusione tra le tecniche classiche: si sfrutta la conoscenza di un oggetto di geometria nota a priori ma anche il principio dell'auto-calibrazione di una scena statica.

4.3.1 Equazioni basilari

Notazioni

Un generico punto 2D lo indichiamo con $m = [u, v]^T$ ed un punto 3D con $M = [X, Y, Z]^T$. Usiamo $\bar{x}$ per denotare il vettore, ottenuto aggiungendo 1 come ultimo elemento: $\bar{m} = [u, v, 1]^T$ e $\bar{M} = [X, Y, Z, 1]^T$. La telecamera viene modellizzata con l'usuale modello *pinhole* (vedi [Appendice A]) : la relazione tra una punto 3D M e la sua proiezione nell'immagine m è data da:

$$s\bar{m} = A[R, t]\bar{M}$$

dove s è un fattore scala e (R, t), chiamati *parametri estrinseci*, sono la rotazione e la traslazione che legano le coordinate *world* alle coordinate del sistema-telecamera; A, ovvero la matrice intrinseca della telecamera, è data dalla matrice:

$$A = \begin{bmatrix} \alpha & \gamma & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{bmatrix}$$

dove (u_0, v_0) sono le coordinate del punto principale, α e β i fattori di scala dovuti agli assi u e v, infine γ è il parametro che descrive l'asimmetria tra i due assi dell'immagine.

Legame tra il piano del modello e la sua immagine

Senza perdita di generalità assumiamo che il piano del modello sia in $Z = 0$ delle coordinate *world*. Denotando le colonne di R con r_i, si può riscrivere **??** come:

$$s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = A \begin{bmatrix} r_1 & r_2 & r_3 & t \end{bmatrix} \begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix} = A \begin{bmatrix} r_1 & r_2 & t \end{bmatrix} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix}$$

Con abuso della notazione, usiamo ancora M per denotare un punto nel piano del modello, $M = [X, Y]^T$ poiché Z = 0. Equivalentemente $\bar{M} = [X, Y, 1]^T$. Di conseguenza un punto M nel modello e la sua immagine m sono legati da una matrice 3x3 H:

$$s\bar{m} = H\bar{M}$$

con $H = A \begin{bmatrix} r_1 & r_2 & t \end{bmatrix}$

Vincoli dei parametri intrinseci

Data un'immagine del piano del modello, può essere stimata una relazione che denotiamo con $H = A \begin{bmatrix} h_1 & h_2 & h_3 \end{bmatrix}$. Possiamo dire quindi che:

$$\begin{bmatrix} h_1 & h_2 & h_3 \end{bmatrix} = \lambda A \begin{bmatrix} r_1 & r_2 & t \end{bmatrix}$$

dove λ è una costante scalare. Sapendo che r_1 e r_2 sono ortonormali, si ha che

$$h_1^T (A^{-1})^T A^{-1} h_2 = 0 \tag{4.1}$$

$$h_1^T (A^{-1})^T A^{-1} h_1 = h_2^T (A^{-1})^T A^{-1} h_2 \tag{4.2}$$

Ci sono due costanti basilari nei parametri intrinseci, dettate dalla relazione tra punto del piano e dell'immagine. Poiché ci sono 6 parametri estrinseci (3 di rotazione e 3 di traslazione) e la relazione implica 8 parametri, se ne possono ottenere solamente 2. Nota che $(A^{-1})^T A^{-1}$ descrive la *conica assoluta* di proiezione dell'immagine. Nel prossimo paragrafo se ne darà una interpretazione geometrica.

Interpretazione geometrica

Non è difficile verificare che il piano del modello, con la notazione proposta, è descritto nel sistema coordinate della telecamera dalle seguenti equazioni:

$$\begin{bmatrix} r_3 \\ r_3^T t \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix} = 0 \tag{4.3}$$

dove $w = 0$ per punti all'infinito e $W = 1$ altrimenti. Questo piano interseca il piano all'infinito in una linea: possiamo veder facilmente che $\begin{bmatrix} r_1 \\ 0 \end{bmatrix}$ e $\begin{bmatrix} r_2 \\ 0 \end{bmatrix}$ sono due punti particolari di essa. Ogni punto appartenente ad essa è quindi combinazione lineare dei due punti, ovvero:

$$x_\infty = a \begin{bmatrix} r_1 \\ 0 \end{bmatrix} + b \begin{bmatrix} r_2 \\ 0 \end{bmatrix} = \begin{bmatrix} ar_1 + br_2 \\ 0 \end{bmatrix} \tag{4.4}$$

Eseguiamo ora l'intersezione tra la riga precedente e la conica assoluta. Per definizione, il punto x_∞, chiamato anche punto circolare, soddisfa $x_\infty^T x_\infty = 0$, ovvero:

$$(ar_1 + br_2)^T (ar_1 + br_2) = 0 \tag{4.5}$$

quindi $a^2 + b^2 = 0$ e la soluzione è $b = \pm ai$ dove $i^2 = -1$. I due punti d'intersezione sono:

$$x_\infty = a \begin{bmatrix} r_1 \pm ir_2 \\ 0 \end{bmatrix} \tag{4.6}$$

La loro proiezione nel piano immagine è data, a parte un fattore scala da

$$\bar{m}_\infty = A(r_1 \pm ir_2) = h_1 \pm ih_2 \tag{4.7}$$

Il punto $\bar{m}_\infty$ è nell'immagine della conica assoluta, descritto da $(A^{-1})^T A^{-1}$. Ciò implica che

$$(h_1 \pm ih_2)^T (A^{-1})^T A^{-1} (h_1 \pm ih_2) = 0 \tag{4.8}$$

Richiedendo che sia la parte reale che immaginaria sia zero si confermano le equazioni precedentemente citate.

4.3.2 Risoluzione del problema di calibrazione

Questo paragrafo fornisce una risposta su come risolvere il problema della calibrazione della telecamera. Si comincerà con una soluzione analitica per poi passare ad un'ottimizzazione con il criterio di *massima-verosimiglianza*. Infine si terrà conto dell'effetto di distorsione delle lenti della telecamera.

Soluzione in forma chiusa

In accordo con [15], sia

$$B = (A^{-1})^T A^{-1} = \begin{bmatrix} B_{11} & B_{12} & B_{13} \\ B_{21} & B_{22} & B_{23} \\ B_{31} & B_{32} & B_{33} \end{bmatrix} =$$

$$= \begin{bmatrix} \frac{1}{\alpha^2} & -\frac{\gamma}{\alpha^2\beta} & \frac{v_0\gamma - u_0\beta}{\alpha^2\beta} \\ -\frac{\gamma}{\alpha^2\beta} & \frac{\gamma^2}{\alpha^2\beta^2} + \frac{1}{\beta^2} & -\frac{\gamma(v_0\gamma - u_0\beta)}{\alpha^2\beta^2} - \frac{v_0}{\beta^2} \\ \frac{v_0\gamma - u_0\beta}{\alpha^2\beta} & -\frac{\gamma(v_0\gamma - u_0\beta)}{\alpha^2\beta^2} - \frac{v_0}{\beta^2} & \frac{(v_0\gamma - u_0\beta)^2}{\alpha^2\beta^2} + \frac{v_0^2}{\beta^2} + 1 \end{bmatrix}$$

Si può notare che B è simmetrica definita da un vettore di sei elementi:

$$b = \begin{bmatrix} B_{11} & B_{12} & B_{22} & B_{13} & B_{23} & B_{33} \end{bmatrix}^T \tag{4.9}$$

Consideriamo la i-esima colonna di H, ovvero $h_i = [h_{i1}, h_{i2}, h_{i3}]^T$, allora si avrà che:

$$h_i^T B h_j = v_{ij}^T b \tag{4.10}$$

con

$$v_{ij} = [h_{i1}h_{j1},\ h_{i1}h_{j2} + h_{i2}h_{j1},\ h_{i2}h_{j2},\ h_{i3}h_{j1} + h_{i1}h_{j3},\ h_{i3}h_{j2} + h_{i2}h_{j3},\ h_{i3}h_{j3}]^T \tag{4.11}$$

di conseguenza, le equazioni (4.6) e (4.7) possono essere riscritte come due equazioni in b:

$$\begin{bmatrix} v_{12}^T \\ (v_{11} - v_{22})^T \end{bmatrix} b = 0 \tag{4.12}$$

Se n immagini del piano modello sono osservate, prendendo altre n equazioni come la precedente, avremmo:

$$Vb = 0$$

dove V è una matrice $2nx6$. Se $n \geq 3$ avremmo in generale una soluzione unica definita da un fattore scala. Se $n = 2$ si po imporre la costante $\gamma = 0$, ovvero aggiungere al set di equazioni quest'ultima: $[0, 1, 0, 0, 0, 0]b = 0$. Se $n = 1$ è possibile risolver solamente due parametri intrinseci della telecamera, ovvero α e β, supponendo di conoscere u_0 e v_0. La soluzione di $Vb = 0$ è conosciuta come autovettore di V^TV associato al più piccolo autovalore. Una volta stimato b si può calcolare tutti i parametri intrinseci della telecamera ed ottenere la matrice A:

$$v_0 = (B_{12}B_{13} - B_{11}B_{13})/(B_{11}B_{22} - B_{12}^2) \tag{4.13}$$

$$\lambda = B_{33} - [B_{13}^2 + v_0(B_{12}B_{13} - B_{11}B_{23})]/B_{11} \tag{4.14}$$

$$\alpha = \sqrt{\lambda/B_{11}} \tag{4.15}$$

$$\beta = \sqrt{\lambda B_{11}/(B_{11}B_{22} - B_{12}^2)} \tag{4.16}$$

$$\gamma = -B_{12}\alpha^2\beta/\gamma \tag{4.17}$$

$$u_0 = \gamma v_0/\beta - B_{13}\alpha^2/\lambda \tag{4.18}$$

Ora che è nota la matrice A,i parametri estrinseci per ogni immagine sono prontamente calcolati.

$$r_1 = \lambda A^{-1}h_1 \tag{4.19}$$

$$r_2 = \lambda A^{-1}h_2 \tag{4.20}$$

$$r_3 = r_1 \chi r_2 \tag{4.21}$$

$$t = \lambda A^{-1}h_3 \tag{4.22}$$

con $\lambda = 1/\|A^{-1}h_1\| = 1/\|A^{-1}h_2\|$. Ovviamente, a causa dei dati rumorosi, la matrice $R = [r_1, r_2, r_3]$ in genere non soddisfa la proprietà di una matrice di rotazione.

Stima alla Massima-Verosimiglianza

La soluzione appena trovata è ottenuta minimizzando una distanza algebrica che non è fisicamente significativa. Si può raffinare il calcolo con l'aiuto della stima a *massima-verosimiglianza*.
Abbiamo n immagini del piano del modello e ci sono m punti in esso. Si assume che i punti nell'immagine siano corrotti da rumori indipendenti e identicamente distribuiti. La stima di *massima-verosimiglianza* si ottiene minimizzando la seguente funzione:

$$\sum_{i=1}^{n}(\sum_{j=1}^{m}(\|m_{ij} - \hat{m}(A, R_i, t_i, M_j)\|^2))$$

dove $\hat{m}(A, R_i, t_i, M_j)$ è la proiezione del punto M_j nell'immagine i. Una rotazione R è parametrizzata da un vettore di 3 parametri, denotato da r, che è parallelo all'asse di rotazione e la sua grandezza è pari all'angolo di rotazione. Per minimizzare la sommatoria si richiede dei valori iniziali di A, $R_i, t_i|i = 1, ...n$.

Rapporti con la distorsione radiale

Fino ad ora non è stata considerata la distorsione dovuta alle lenti della telecamera. Solitamente nelle telecamere è significativa la distorsione radiale. in questo paragrafo si cercherà dunque di darne un significato nel modello.
Siano (u, v) le coordinate ideali di un pixel nell'immagine e $(\breve{u}, \breve{v})$ le coordinate reali dell'immagine osservata. I punti ideali sono la proiezione dei punti della matrice di pixel in accordo con il modello *pinhole*. Quindi (x, y) e $(\breve{x}, \breve{y})$ sono le coordinate ideali e reali dell'immagine. Avremo dunque:

$$\breve{x} = x + x[k_1(x^2 + y^2) + k_2(x^2 + y^2)^2]$$
$$\breve{y} = y + y[k_1(x^2 + y^2) + k_2(x^2 + y^2)^2]$$

dove k_1 e k_2 sono i coefficienti della distorsione radiale. Il centro della distorsione radiale è lo stesso del punto principale. Dalle formule $\breve{u} = u_0 + \alpha\breve{x} + \gamma\breve{y}$ e $\breve{v} = v_0 + \beta\breve{y}$, assumendo $\gamma = 0$ si ricava:

$$\breve{u} = u + (u - u_0)[k_1(x^2 + y^2) + k_2(x^2 + y^2)^2]$$

$$\check{v} = v + (v - v_0)[k_1(x^2 + y^2) + k_2(x^2 + y^2)^2]$$

Stima di Massima Verosimiglianza completa

Una estensione della stima citata precedentemente consiste nel ricercare tutti i parametri minimizzando la seguente sommatoria:

$$\sum_{i=1}^{n}(\sum_{j=1}^{m}(\|m_{ij} - \hat{m}(A, k_1, k_2, R_i, t_i, M_j)\|^2))$$

dove $\hat{m}(A, k_1, k_2, R_i, t_i, M_j)\|^2)$ è la proiezione del punto M_j nell'immagine i. Ancora una volta questo è un problema non-lineare di minimizzazione.

4.3.3 Operazione di calibrazione

La procedura di calibrazione eseguita in laboratorio risente di alcuni accorgimenti rispetto la versione standard delle telecamere RGB o B/N.

Figura 4.5: sistema di fissaggio della matrice di acquisizione

Figura 4.6: illuminatore alogeno da 180 Watt

La telecamera infrarossi del $Kinect^{TM}$ infatti capta solamente una parte dello spettro luminoso solare ed il led IR presente interferisce alla procedura di calibrazione. Per la fase di calibrazione è stato disattivato via software l'accensione

del led IR ed è stato utilizzato un illuminatore artificiale composto da 3 lampade alogene di potenza *60 Watt*. L'illuminazione artificiale aggiunta comprende lo spettro dell'infrarosso e permette al sensore IR di rilevare le immagini con miglior rapporto segnale-rumore. Come trama è stata scelta una scacchiera bianca-nera stampata su carta e fissata su una lastra in plexiglass in modo da renderla piana e rigida. Sono state acquisite le immagini e presi i punti di intersezione tra le varie celle della trama stimando poi i parametri e ricavando la matrice A:

$$A = \begin{bmatrix} 587.346 & 0 & 318.224 \\ 0 & 587.291 & 236.393 \\ 0 & 0 & 1 \end{bmatrix}$$

4.4 Ricostruzione grafica

Una volta memorizzate le immagini *depth* ed *RGB* con le relative posizioni ed orientazioni è possibile eseguire una ricostruzione grafica 3D della scena ricorrendo alla trasformazione dettata dal modello *pin-hole* della telecamera. Fondamentale in questa fase è la matrice dei parametri intrinseci della telecamera calcolata al paragrafo precedente.

4.4.1 Supporto Kinect e Matrice di rototraslazione Kinect-Robot

Tramite il software $SolidWorks^R$ è stato realizzato il supporto per $Microsoft\ Kinect^{TM}$. Ridisegnando il dispositivo è stato possibile ottenere un supporto che rispettasse fedelmente le dimensioni e i ganci di attacco sulla base senza bloccare la movimentazione di tilt del $Kinect^{TM}$. Via software è stato possibile determinare efficacemente la matrice di rototraslazione tra la flangia Robot e la telecamera IR che poi sarà utilizzata nei calcoli di ricostruzione grafica:

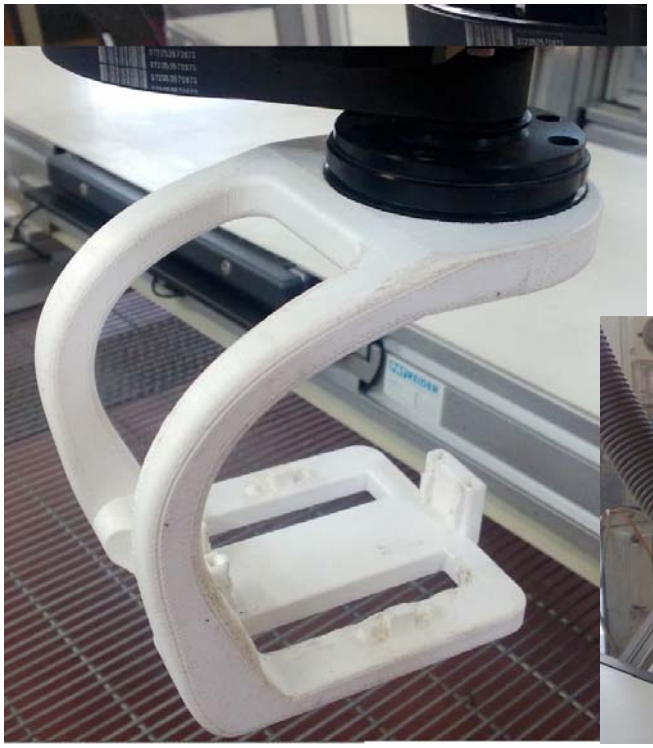


Figura 4.7: Supporto per il $Microsoft\ Kinect^{TM}$.

Figura 4.8: $Microsoft\ Kinect^{TM}$ fissato alla flangia del robot in fase di aquisizione.

$$T_{Robot-Kinect} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & +12 \\ 0 & 0 & 1 & -51 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.23}$$

4.4.2 Riduzione rumore

Un tradizionale sistema di acquisizione digitale comprende un sottocircuito di natura analogica. La prima fase del processo di acquisizione infatti consiste nella trasformazione della sorgente luminosa sotto forma di fotoni in una carica elettrica che dovrà essere trasformata in un valore di tensione per poi essere espressa in una grandezza binaria. In questi casi il dato è influenzato da un rumore Gaussiano, ovvero un rumore a media nulla con spettro costante in tutte le frequenze. Si consideri ora la matrice di profondità che il $Kinect^{TM}$ fornisce ad ogni acquisizione. Il valore di ogni cella di questa matrice è frutto di un algoritmo di *spatial-multiplexing* (vedi cap.1); il rumore generato in un punto non si può ipotizzare Gaussiano poiché frutto di un sistema di calcolo complesso e, in parte, sconosciuto.

Un'analisi dei dati acquisiti a dispositivo fermo dimostra che i dati appaiono disturbati nei primi 30s assestandosi ad un valore finale. Altro fatto da constatare: le acquisizioni eseguite con la presenza di luce solare implicano del rumore impulsivo, ovvero punti isolati in cui il valore *depth* non è espresso. A fronte di ciò è stato rilevato un miglioramento nell'acquisizione di profondità in assenza di fonti luminose e temporizzando l'inizio dell'acquisizione ad ogni posizionamento.

Il filtraggio eseguito nelle immagini esegue una cancellazione dei valori *depth* troppo dispersivi nel tempo. Terminate le acquisizioni in un posizionamento, viene calcolata la media temporale di ogni elemento ij-esimo ed eliminati in caso la varianza superi il limite massimo di varianza σ^2_{MAX}. L'eliminazione di punti profondità rumorosi ha come conseguenza una ricostruzione 3D meno densa, tuttavia, un maggior numero di punti di acquisizione consente di ovviare al problema.

Definite K acquisizioni per ogni posizione e d_{ij}^k il valore di profondità del pixel ij nel frame k, definiamo il valore atteso dell'elemento ij-esimo:

$$E[d_{ij}] \cong \frac{\sum_{k=1}^{K}(d_{ij}^k)}{K}$$

e la varianza sarà $\sigma^2(d_{ij}) = E[(d_{ij} - E[d_{ij}])^2]$. Quindi, la condizione da rispettare sarà: $\sigma^2(d_{ij}) \leq \sigma_{MAX}^2$ Per tutti gli elementi della matrice che rispettano la condizione sopra riportata viene eseguito in filtraggio in media temporale:

$$\bar{d}_{ij} = \frac{\sum_{k=1}^{K}(d_{ij}^k)}{K}$$

4.4.3 Ricostruzione

Una volta eseguita l'operazione di filtraggio dei dati, dalle K matrici $D_n^k|k = 1, ...K$ di profondità della posizione n, si ottiene un unica matrice, che definiremo $\hat{D}_n$. Queste N matrici compongono N differenti nuvole di punti che, fuse insieme, andranno a ricostruire la scena.
Il primo passo consiste nello sfruttare il modello *pinhole* della telecamera (vedi Appendice A) per trovare la matrice di traslazione dal centro focale della telecamera IR del $Kinect$ alla terna di ogni singolo punto. Siano δx, δy, δz le coordinate del punto P_{Kinect} relativo alla cella ij di una matrice *depth* riferite alla telecamera IR e d_{ij}^n il valore di profondità, tenuto conto della matrice A calcolata in fase di calibrazione:

$$\delta x = d_{ij}^n$$

$$\delta y = d_{ij}^n(j - 318.224)/587.346$$

$$\delta z = d_{ij}^n(236.393 - i)/587.291$$

Definito il punto nello spazio $P_{Kinect} = \begin{bmatrix} \delta x \\ \delta y \\ \delta z \\ 1 \end{bmatrix}$ in coordinate $Kinect$ si ricava il corrispettivo P_O rispetto alla terna di riferimento O con la seguente trasformazione:

$$P_{Robot} = T_{Robot-0} T_{Kinect-Robot} P_{Kinect} =$$

$$= \begin{bmatrix} \cos(\gamma) & -\sin(\gamma) & 0 & x_{end} \\ \sin(\gamma) & \cos(\gamma) & 0 & y_{end} \\ 0 & 0 & 1 & z_{end} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & +12 \\ 0 & 0 & 1 & -51 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \delta x \\ \delta y \\ \delta z \\ 1 \end{bmatrix}$$

Eseguito il calcolo per tutti i pixel della matrice di profondità del $Kinect^{TM}$ che hanno avuto esito positivo si ottiene un insieme di punti H_n con un numero massimo teorico di $640x480$ elementi per ogni $n\epsilon[0, ..N]$.

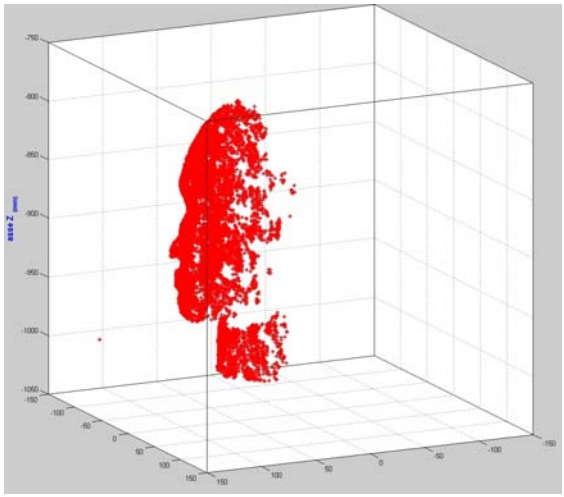


Figura 4.9: nuvola di punti - singola acquisizione di una testa artificiale

Nella prima figura un acquisizione singola della testa, eseguita alla distanza media di 50 cm, fornisce una fitta nuvola di punti. Nella seconda figura, quattro acquisizioni eseguite con uno sfasamento di 90°. In questo caso la scena è molto più ampia e la densità di $pixel/mm^3$ è inferiore, anche a causa della riflessività

dell'alluminio, materiale di cui è composta la cella. Grazie all'acquisizione da più punti di vista è possibile ottenere una nuvola di punti sempre più densa e si rilevano le zone non raggiunte in precedenza.
Al termine del processo di acquisizione si otterrà un insieme H dei punti acquisiti e filtrati:

$$H = H_1 \cup H_2 \cup H_3 \cup ...H_N$$

4.4.4 Problematiche

Il metodo di ricostruzione proposto in questa prima parte richiede di sapere posizione ed orientazione della telecamera con precisione assoluta. Solitamente in un sistema con telecamera fissa la sua posizione può essere determinata sfruttando una matrice di calibrazione che abbia posizione e dimensioni note. Tuttavia il caso in questione è diverso. L'installazione della telecamera in un apparato mobile richiede che quest'ultimo, ovvero il robot *Adept Quattro* fornisca posizioni ed orientazioni reali dell'ultimo membro, ovvero la flangia su cui è fissato il supporto del $Kinect^{TM}$. Gli encoder installati nel robot si trovano a monte della catena cinematica, e la disposizione degli arti in parallelo fa si che l'errore sul posizionamento dell'end effector sia superiore rispetto ai robot scara o antropomorfi. Da quanto riportato nel manuale *Adept* la Oltre a ciò è importante considerare le caratteristiche meccaniche del sistema *supporto-Kinect*. Ipotizzato che la deformazione del supporto sia irrilevante e che il $Kinect^{TM}$ sia sempre in perfetta orizzontalità grazie al loop di controllo del sistema di regolazione tilt, il dispositivo Microsoft detiene un gioco rotazionale non quantificato nell'asse z (dovuto alla meccanica del *tilt*).

Ciò si traduce in una errata conoscienza della posizione ed orientazione della telecamera con conseguente spostamento della nuvola di punti. La ricostruzione formata da più nuvole di punti appare discontinua causa il *matching* non corretto. Per ovviare a ciò esistono in letteratura innumerevoli metodologie. In questo contesto viene proposto una versione ottimizzata del cosidetto algoritmo *ICP* (Itertive Closest Point.

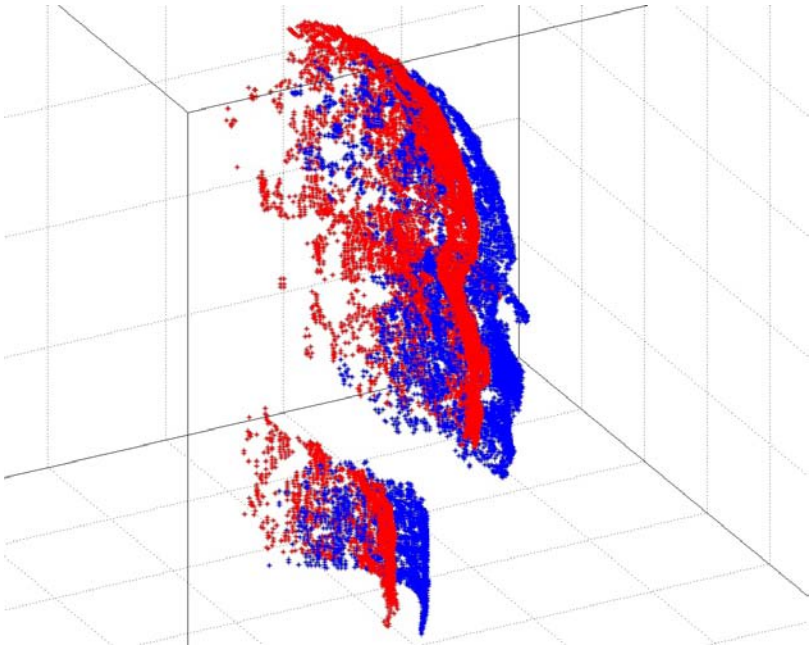


Figura 4.10: accoppiamento errato tra nuvole di punti

4.4.5 Matching tra nuvole di punti

Algoritmo ICP

Iterative Closest Point, o **ICP** è un algoritmo impiegato per minimizzare la differenza tra due nuvole di punti. Spesso viene utilizzato anche per la ricostruzione di superfici 2D o 3D a partire da differenti scansioni. La semplicità dell'algoritmo permette la sua implementazione in sistemi real-time, tuttavia si tratta di un metodo iterativo: talvolta solo più trasformazioni permettono di ottenere il risultato dovuto.
Sostanzialmente l'algoritmo necessita di aver a disposizione le due nuvole di punti ed il numero di iterazioni da effettuare. Il processo si suddivide nelle seguenti fasi:

- associare i punti attraverso il **NNS** (Nearest Neighbor Criteria): per ogni punto in una nuvola di punti viene individuato il punto più vicino nella seconda nuvola;
- stimare i parametri per la trasformazione: verrà effettuata una traslazione ed una rotazione determinata da una *stima ai minimi quadrati*;
- trasformare i punti grazie ai parametri appena individuati;

- successiva iterazione (con eventuale condizione per bloccare l'iterazione successiva).

La differenziazione tra i vari algoritmi *ICP* sta sopratutto nel codice utilizzato per il **NNS**. In questo ambito infatti sono molteplici le varianti presenti in letteratura.
Nel codice proposto per ogni punto della matrice P_{ICP} dei punti di riferimento viene calcolata la distanza euclidea con tutti i punti appartenenti ad un'altra acquisizione (matrice P_n), stipulandone i K punti più vicini. Si ottiene dunque una lista dei punti corrispondenti più vicini con le relative distanze.

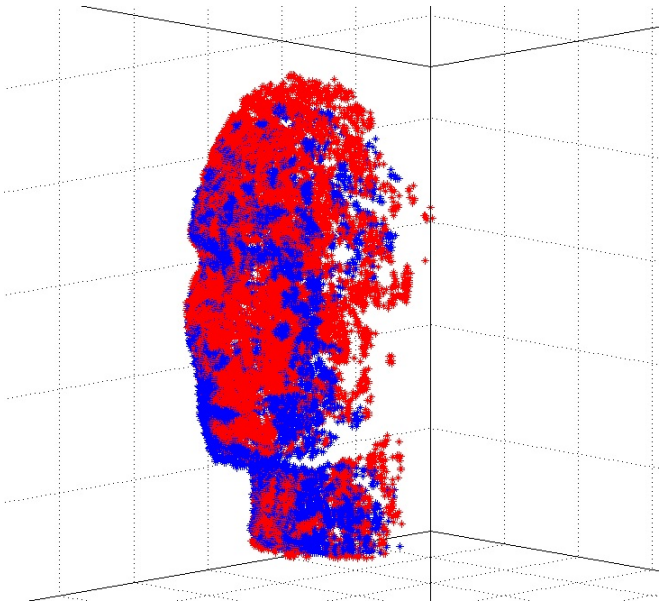


Figura 4.11: matching tra due acquisizioni da due posizioni diverse.

Il problema dell'accoppiamento di due nuvole di punti si risolve minimizzando la somma delle distanze dei punti della prima nuvola rispetto a quelli della seconda. Nell'algoritmo proposto, denominati P_{n-r} i R punti della seconda nuvola e P_{ICP-rk} i corrispettivi k vicini, ciò si risolve cercando il minimo del fattore

$$D_{TOT} = \sum_{r=1}^{R} \sum_{k=1}^{K} \|P_{n-r} - P_{ICP-rk}\|$$

proporzionale alla distanza tra le due nuvole.
Ad ogni iterazione si cerca di trovare la traslazione e la rotazione necessaria per

ottenere una distanza somma minima definita con D_{min}, se l'operazione è possibile senza raggiungere il limite massimo di iterazioni $Iter_{MAX}$ è blocca l'esecuzione del ciclo.
Il calcolo della rotazione e della traslazione può essere effettuato in maniera indipendente.

Le tre dimensioni che implicano la traslazione vengono calcolate direttamente al vettore tridimensionale

$$T_{Titer} = -\frac{\sum_{r=1}^{R}\sum_{k=1}^{K}(P_{n-r} - P_{ICP-rk})}{KR}$$

Grossomodo la loro azione permette di ottenere una sovrapposizione tra i baricentri di due differenti nuvole.
Il secondo passo di ogni iterazione consiste nell'individuare i tre angoli di rotazione α, β, γ nei corrispettivi assi x, y, z. Per i punti appartenenti ad ogni nuvola viene calcolata la normale alla superficie formata dalla triangolazione con 2 i vicini scelti. Dato quindi un normale del punto i-esimo,

$$\vec{n^i} = [n_x^i, n_y^i, n_z^i]^T$$

si ricavano i suoi angoli di orientazione rispetto la terna di riferimento *mondo*:

$$\alpha^i = \arccos(n_x^i)$$

$$\beta^i = \arccos(n_y^i)$$

$$\gamma^i = \arccos(n_z^i)$$

Avendo dunque a disposizione l'informazione dei punti di due nuvole, è possibile calcolare gli angoli di rotazione α, β, γ semplicemente come differenza degli angoli risultanti da:

$$\alpha = \bar{\alpha}_{ICP} - \bar{\alpha}_2$$

$$\beta = \bar{\beta}_{ICP} - \bar{\beta}_2$$

$$\gamma = \bar{\gamma}_{ICP} - \bar{\gamma}_2$$

La matrice di rotazione T_{Riter} sarà definita come prodotto delle tre matrici:

$$T_{Riter} = T_{\alpha_{iter}} T_{\beta_{iter}} T_{\gamma_{iter}}$$

La rototraslazione totale per portare la seconda nuvola di punti ad accostarsi alla prima sarà data dalle matrici T_R e T_T, rifinite nelle I iterazioni successive dell'algoritmo:

$$T_R = \prod_{iter=1}^{I} T_{Riter}$$

$$T_T = \sum_{iter=1}^{I} T_{Titer}$$

Si allinea dunque la seconda nuvola di punti nel seguente modo:

$$P_{2mod} = T_R P_2 + T_T$$

Algoritmo di ricostruzione con ICP

La ricostruzione richiede di eseguire ripetutamente l'algoritmo ICP per ogni nuvola che dovrà essere accostata alla precedete. Eseguire l'accoppiamento tra due superfici può essere dispendioso, tuttavia, conosciute le porzioni di superficie coperte da entrambe le nuvole di punti, è possibile limitare l'elaborazione a queste, evitando di inceppare in errori dovuti a altri possibili accostamenti. L'algoritmo proposto è stato quindi ottimizzato per l'acquisizione di un oggetto presente nella parte centrale della scena tramite un movimento circolare attorno ad esso.

La nuvola di punti generata dalla prima immagine costituisce il riferimento per la prima iterazione e viene inserita nella matrice P_{ICP} a partire dai punti generati dalla colonna di pixel a destra alla sinistra, in modo da esser concordi alla direzione delle future acquisizioni. Viene generata quindi la matrice P_2 contenente la nuvola di punti rilevata alla seconda posizione. L'euristica dell'algoritmo consiste in una taratura di un parametro q che deve indicare grossomodo la percentuale di punti di ognuna delle due matrici che sono compresi nel volume selezionato per la ricostruzione (esso non è dipendente dal numero di punti rilevati ad ogni acquisizione). Le matrici P_{ICP-dx} e P_{2-sx} saranno dunque argomento dell'algoritmo **ICP**, la trasformazione dettata dalle matrici T_R e T_T permette il corretto

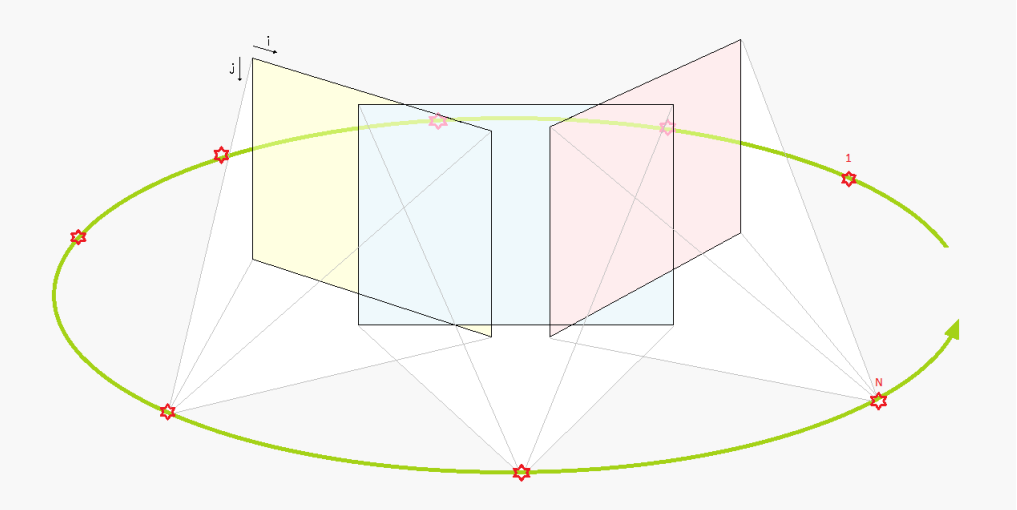


Figura 4.12: rappresentazione della metodologia di acquisizione.

allineamento dei punti della matrice P_2 al riferimento P_{ICP}. Infine un'unione dei punti delle due matrici amplia la nuvola consolidata. Ricorsivamente si esegue l'algoritmo con una nuova posizione di acquisizione fino ad ottenere una matrice P_{ICP} tale che:

$$dim(P_{ICP}) = \sum_{n=1}^{N}(dim(P_n))$$

Per cui tutte le immagini saranno espresse in punti nello spazio formanti la superficie dell'oggetto inquadrato dalla scena.

4.4.6 Selezione dei punti e conversione a Voxel

Quando si effettuano acquisizioni con un numero elevato di immagini di profondità e le superfici inquadrate sono di varia natura e/o distanza si rischia di immagazzinare un'elevata mole di dati, talvolta senza utilità. Date le prerogative iniziali di questa ricerca e conosciuti i limiti del $Kinect^{TM}$ si rende necessaria una selezione dei punti trovati.

Una prima scelta viene eseguita eliminando tutti i punti non appartenenti al volume della cella: le pareti di plexiglass sono trasparenti, di conseguenza il sistema di visione rileva punti al di fuori della cella. Considerato

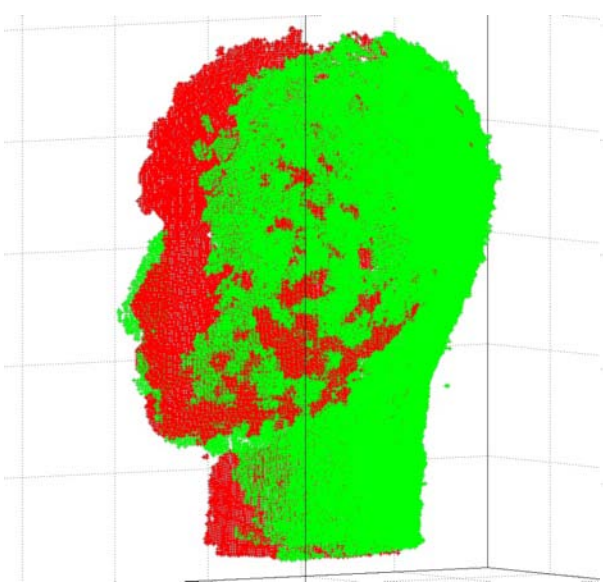


Figura 4.13: fase di ricostruzione: matching tra un riferimento composto dalla prima acquisizione (rosso) ed altre tre (verde).

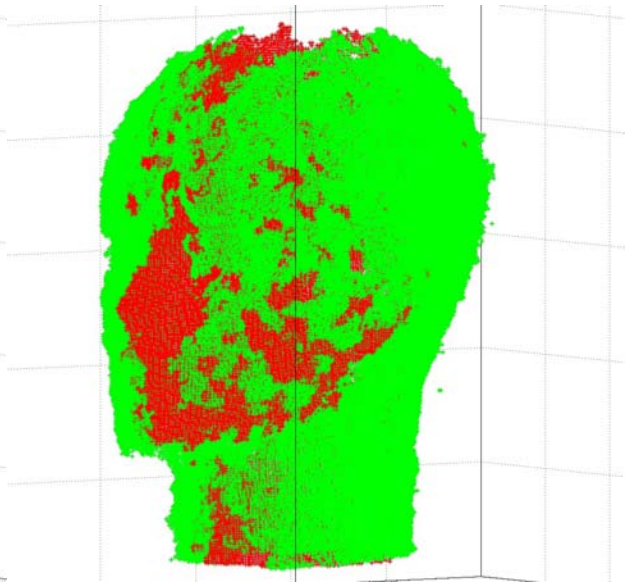


Figura 4.14: ricostruzione completa della testa da otto acquisizioni tramite algoritmo ICP.

$$C = \{x\epsilon\Re| -1650 < x < 1650;\ y\epsilon\Re| -1650 < y < 1650;\ z\epsilon\Re| -1350 < z < 0\}$$

l'insieme dei punti appartenenti alla cella ed al suo interno, si ottiene:

$$H^I = H \cap C$$

In alcune zone la densità di pixel in una superficie venutasi a creare è elevata e superiore alla risoluzione del sistema di visione. Per questo motivo lo spazio tridimensionale è stato diviso in cubi di lato $2mm$, definiti nel campo grafico col termine *voxel*. Il primo punto a determinare un *voxel* è dominante: qualsiasi altro punto interno al volume viene scartato. Con questo algoritmo si riesce a contenere la complessità computazionale di un successivo lavoro di *meshing* o di visualizzazione grafica preservandone la qualità; si ricava dunque un'insieme di dimensioni inferiori

$$H^{II} \subseteq H^I$$

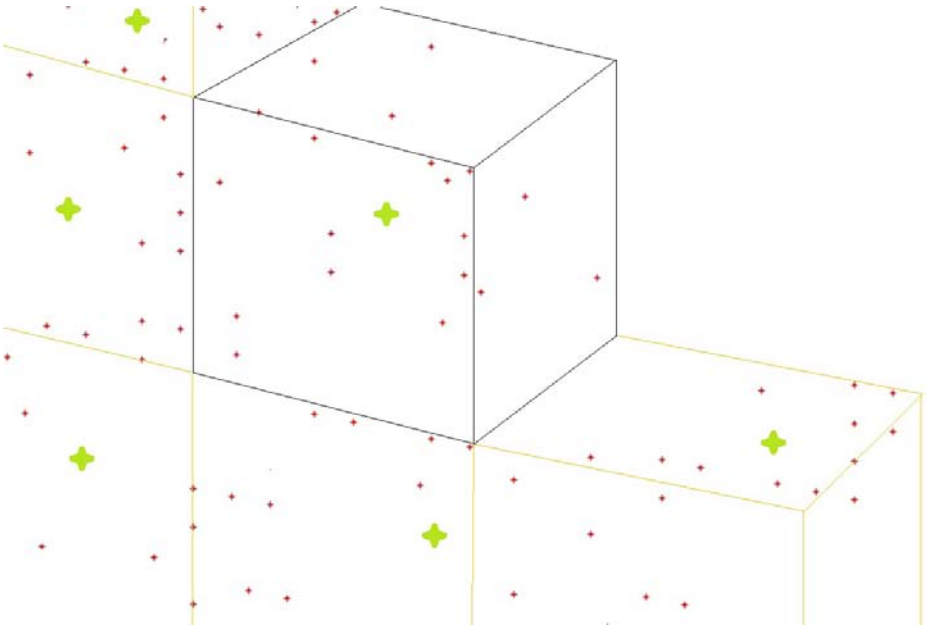


Figura 4.15: Eliminazione dei punti superflui all'interno di un voxel.

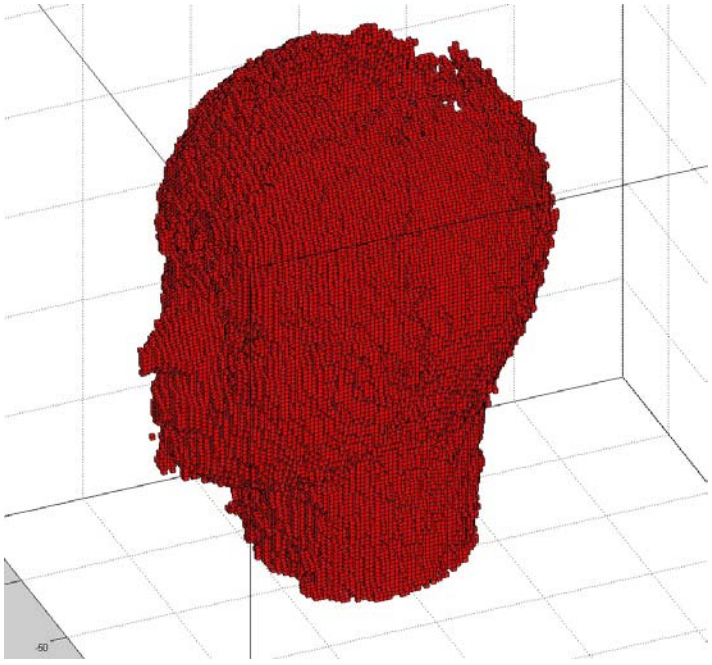


Figura 4.16: trasformazione in voxel - testa

4.5 Analisi della ricostruzione

Al fine di valutare la qualità di una ricostruzione grafica del $Microsoft\ Kinect^{TM}$ si è scelto di confrontare la scansione della testa di gomma ottenuta in assenza di luce. Presso il laboratorio di Metrologia Geometrica ed Industriale del **DIMEG** è stata eseguita una ricostruzione accurata della testa tramite il *Faro ScanArm* per poi eseguire un confronto con il software $Geomagic^{R}$ del modello generato dalle scansioni da parte del sistema Adept Quattro - Kinect.

4.5.1 Faro ScanArm

Il dispositivo ScanArm della Faro è un sistema di misurazione portatile che può agire sia per contatto sia per scansione laser. Viene utilizzato principalmente per il *reverse engineering*. Esso è costituito da una pistola munita di tastatore e telecamera 3D a triangolazione laser connessa ad un'articolazione artificiale a 7 *gradi di libertà*. Date le dimensioni dei vari membri del braccio mobile esso è adatto alle scansioni di oggetti di ridotte dimensioni.

Figura 4.17: Faro ScanArm

Il sistema ha bisogno di una taratura prima dell'acquisizione di un oggetto, sfrutta quindi un punto 3D rilevato in questo procedimento come coordinata di riferimento. La ricostruzione è possibile grazie alla cinematica inversa. Ogni giunto permette un *grado di libertà* ed è provvisto di *encoder*, ciò consente di saper sempre la posizione e l'orientazione dell'ultimo membro della catena cinematica, da qui la ricostruzione.
Dal metodo della triangolazione vengono generate linee di 640 *pixel* che verranno convertite direttamente in punti nel sistema tridimensionale.
Supponendo la temperatura ambiente costante il dispositivo vanta di un'accuratezza variabile da 0.024 *mm* a 0.064 *mm* dipendentemente dalla posizione, per questo motivo questo dispositivo è stato utilizzato come strumento di verifica della qualità delle scansioni del $Kinect^{TM}$, assumendo che il modello da esso creato sia fedele all'oggetto reale.

4.5.2 Geomagic Qualify

Geomagic Qualify è un software di analisi e ricostruzione di superfici e volumi. Esso permette di gestire nuvole di punti, file grafici ottenuti dalla triangolazione, eseguire *matching* ed allineamento tra superfici. Oltre a ciò è possibile implementare filtri grafici per la riduzione del rumore, il lisciamento di superfici e sono presenti algoritmi per la ricostruzione di zone parzialmente note. Il software è stato utilizzato per eseguire un confronto grafico ed un'analisi quantitativa tra le ricostruzioni effettuate nel periodo di tesi.

4.5.3 Scansione e confronto

In una prima fase si è eseguita la scansione della testa tramite il sistema appena citato. L'utilizzo della tecnica di triangolazione in una camera 3D con fascio luminoso di tipologia *laser* non consente un'immunità alle variazioni cromatiche delle superfici: la potenza luminosa del fascio *laser* va quindi adattata alla tipologia ed al colore superficie interessata. Nel caso il fascio *laser* interessi superfici di vario genere e colore è inevitabile la presenza di rumore nella lettura. Durante la prova

Figura 4.18: immagine reale della testa

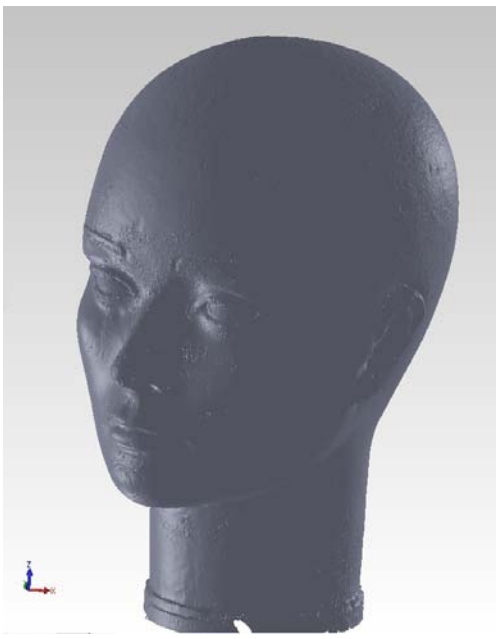


Figura 4.19: testa acquisita con Faro ScanArm e Geomagic

sono stati riscontrati problemi nell'acquisizione delle zone colorate in nero, le quali hanno subito più volte la scannerizzazione al fine di infittire la nuvola di punti che appariva in principio rada. Un effetto evidente della variazione cromatica è stato riscontrato nella sopracciglia: mentre nella realtà esse son solo colorate, nel modello 3D derivato esse appaiono in rilevo.

Il processo di acquisizione produce una nuvola di punti di elevata densità che sarà poi trattata col software *Geomagic Qualify* in modo da determinare nuovi punti appartenenti alle micro-zone in cui la triangolazione non ha trovato corrispondenza. I metodi di ricostruzione grafica possono sfruttare congiunzioni planari oppure curve di raccordo che seguono le geometrie circostanti alla zona critica. Successivamente si applica l'algoritmo di riduzione della nuvola di punti secondo criteri di distanza superficiale tra i punti che compongono la superficie. L'esigenza è dunque fornire una superficie più accurata e densa di quella ottenuta col $Kinect^{TM}$, per cui una distanza di 0.5 mm tra punto-punto permette di ottenere un *meshing* ottimale a fronte della forma reale dell'oggetto in questione. In questo modo è stato ottenuto un modello tridimensionale della testa che poi verrà utilizzato come riferimento per il confronto con le acquisizione svolte nella

cella robotizzata.
Sfruttando i *tool* del software è possibile trasformare la nuvola di punti ottenuta dal sistema Adept-Kinect in una superficie per poi esser allineata, con algoritmi ricorsivi, alla superficie del modello di riferimento (ottenuto con la scannerizzazione laser). Il metodo ottimizza in base alla distanza tra le due superfici. Per la ricostruzione grafica sono state utilizzate nove acquisizioni (filtrate dal rumore) effettuate ad una distanza media di 50 *cm* durante un moto di rivoluzione attorno ad essa. Notasi che essendo questa la distanza minima per una corretta acquisizione la risoluzione risulta la massima possibile: nel piano bidimensionale x-y si ha teoricamente una risoluzione di circa 1 *mm*, mentre per il valore *depth* si consta una risoluzione pari a 1.6 *mm* a fronte di un'accuratezza di 10 *mm* [3]. Una prima analisi evidenzia la presenza di rumore nelle immagini di profondità del $Kinect^{TM}$ che appaiono con effetto granulato di una superficie realmente liscia.
Al fine di testare separatamente la qualità delle ricostruzioni grafiche del $Kinect^{TM}$ e la bontà del metodo ICP sviluppato sono state eseguite due prove distinte.
Nella prima prova si esegue un *matching* tra le nove immagini (proiettate nello spazio 3D) direttamente con la funzione *allignement* di *Geomagic*. Si accostano dunque le nuvole di punti generate dalle singole acquisizioni nel modello vero. Tuttavia questo non si tratta di un metodo completamente automatico: per effettuare l'accoppiamento delle varie parti si richiede che l'operatore evidenzi le zone in cui eseguire l'algoritmo ICP stabilendo 3 punti di riferimento per ogni nuvola. Si riscontrano deformazioni fino a 5 *mm* nella ricostruzione di un oggetto di diametro 180 *mm* alla distanza minima per un'acquisizione del $Kinect^{TM}$ (in *near-mode*) ovvero 40-50 *cm*. Nella figura 4.22 si notano in blu le zone in cui la ricostruzione si discosta in negativo rispetto il modello ed in giallo in senso positivo, ovvero verso l'esterno. La deviazione standard è di 1.425 *mm*; il software calcola questo valore prendendo come riferimento la zona in cui il *meshing* ha dato esito positivo. La deviazione standard viene calcolata prendendo la distanza minima tra ogni punto dell'immagine da confrontare e il più vicino punto nel modello di riferimento 4.20 [23].

Una seconda analisi (vedi 4.23) viene effettuata tra il modello di riferimento

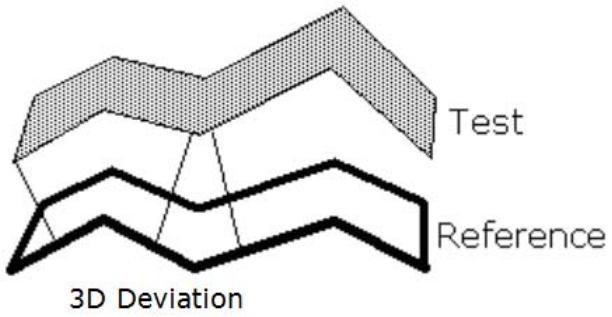


Figura 4.20: scelta delle deviazioni da parte di Geomagic Qaulify

	matching Geomagic	algoritmo ICP
Deviazione MAX + [mm]	8,217	16,754
Deviazione MAX - [mm]	10,272	9,981
Deviazione Media + [mm]	0,670	2,745
Deviazione Media - [mm]	1,496	2,645
Deviazione Standard [mm]	1,425	3,361

Figura 4.21: analisi della ricostruzioni effettuate

e la ricostruzione eseguita dal software *Matlab* con l'algoritmo di ricostruzione sviluppato durante il periodo di tesi. I risultati ottenuti sono paragonati nella tabella di 4.21. La modalità quasi completamente automatica dell'algoritmo non permette di ottenere risultati migliori rispetto al *matching* manuale. Talvolta infatti la similitudine tra due nuvole di punti corrispondenti a due porzioni di superficie diverse nell'oggetto reale può causare un allineamento non idoneo delle medesime.

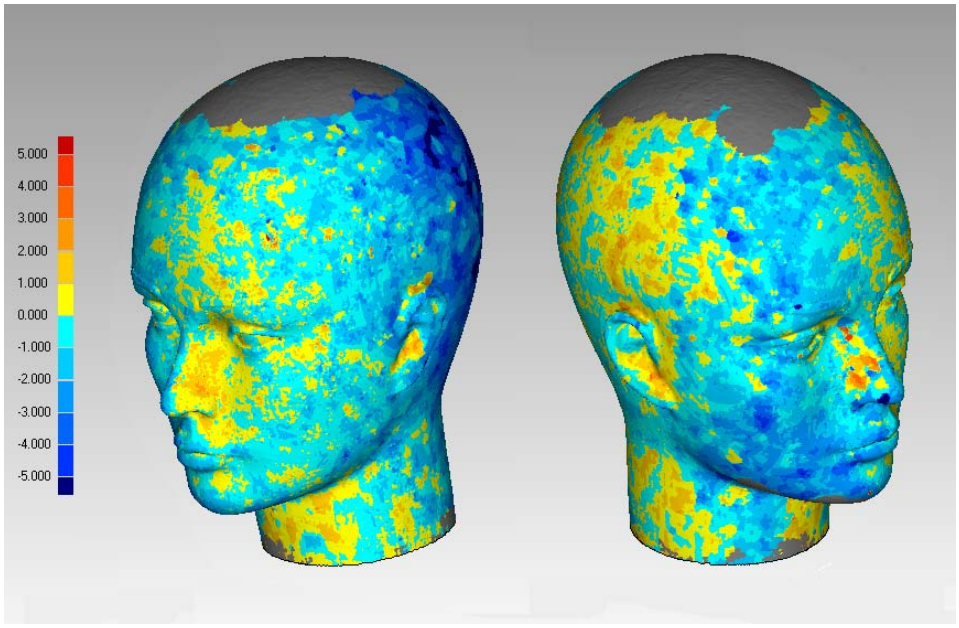


Figura 4.22: analisi della ricostruzione con immagine del Kinect e matching effettuato da Geomagic Qaulify

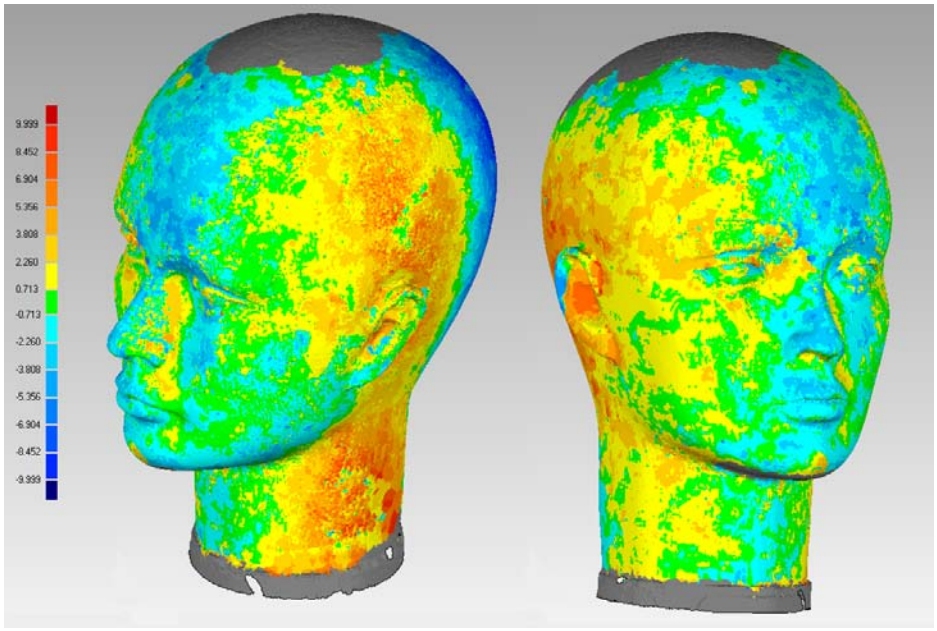


Figura 4.23: analisi della ricostruzione con immagine del Kinect e matching effettuato dall'algoritmo sviluppato

Conclusioni

L'applicazione delle odierne tecnologie nell'automazione ha portato ad un considerevole utilizzo della visione 3D nella rilevazione di geometrie ed oggetti.
Le prove effettuate hanno dimostrato i limiti dell'uso del $Kinect^{TM}$ nell'ambito industriale. Da un lato un alto frame-rate permette acquisizioni di scene in movimento, dall'altro la bassa risoluzione influisce sulla qualita' della ricostruzione. Le euristiche interne del dispositivo tendono a fornire un'immagine priva di rumore ma compromettono talvolta l'offset nelle tre dimensioni dell'immagine. E' stato analizzato dunque un algoritmo di accoppiamento tra le varie nuvole di punti per ovviare il problema. Cio' permette un corretto allineamento tra le nuvole a discapito dell'effettivo posizionamento rispetto le coordinate *world* reali.
Il confronto con una ricostruzione eseguita da un braccio di scansione professionale ha permesso di verificare una buona qualita' delle ricostruzioni cosi ottenute.

Appendice A

Modello Pin-hole di una telecamera

Consideriamo un sistema tridimensionale con gli assi x, y, z e origine nel punto **O**, che chiameremo centro della proiezione. Chiameremo poi piano **S** il piano parallelo al piano generato dagli assi x e y nella coordinata negativa *-f*, dove f è un un parametro intrinseco che sta ad indicare la lunghezza focale della telecamera. Imponiamo per i prossimi calcoli la convezione della mano destra per l'orientazione degli assi. Prendendo il sistema di riferimento 2D associato al sensore:

$$\begin{aligned} u &= x + c_x \\ v &= y + c_y \end{aligned} \tag{A.1}$$

che chiameremo S_{2D}, l'intersezione c dell'asse z con il piano del sensore avrà coordinate $[u = c_x, v = c_y]^T$. Il sensore sarà costituito da una matrice mxn di celle fotosensibili, ognuna delle quali farà riferimento ad un singolo pixel dell'immagine catturata. Ad ognuno dei pixel del sensore corrisponderà dunque una coordinata $p = [u, v]^T$ da cui si ricava un punto P di un oggetto nello spazio.

Immediato ricavare dalle relazioni precedenti che

$$\begin{aligned} u + c_x &= f\frac{x}{z} \\ v + c_y &= f\frac{y}{z} \end{aligned} \tag{A.2}$$

Dove f viene intesa la coordinate negativa di modulo la lunghezza focale. Le relazioni descritte qui sopra sono del tutto corrette se si considera che l'immagine venga catturata da un sistema *pin-hole*, ovvero con un sottile foro posizionato nell'origine degli assi **O**. La luce in questo caso sarebbe costretta a passare su di esso e da qui facile intuire le proiezioni dell'immagine. Tuttavia, per delle caratteristiche dovute ai sistemi per la cattura di immagini, è più pratico l'utilizzo di ottiche formate da serie di lenti focali piuttosto che di un foro. Anche in questo caso è possibile ricondursi ad ogni punto P semplicemente sfruttando il modello *pin-hole* se il centro ottico è in **O** e l'asse z è perfettamente ortogonale al sensore presente nella telecamera e costituisce l'asse dell'ottica utilizzata.

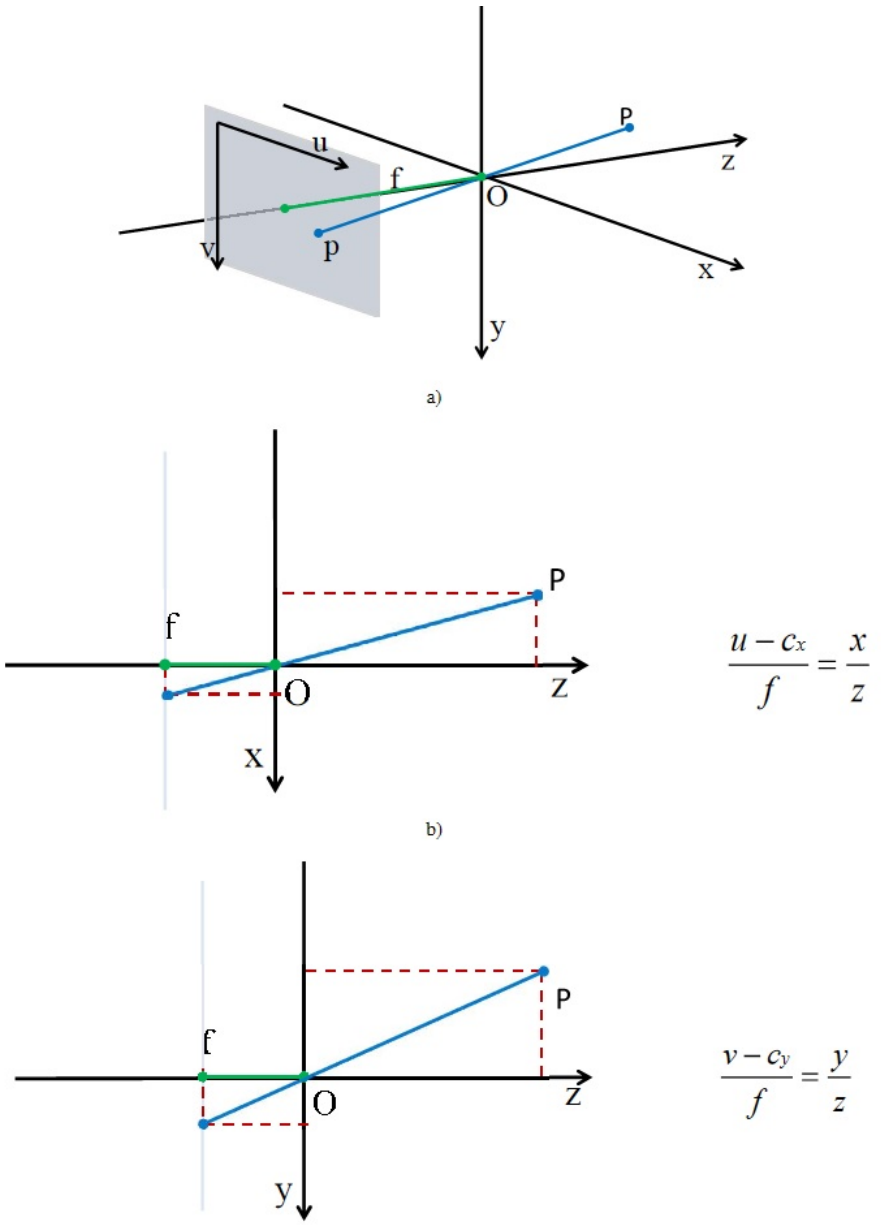


Figura A.1: a) Proiezione tridimensionale del punto P b) Sezione orizzontale c) Sezione verticale

Appendice B

Parametri intrinseci ed estrinseci di una telecamera

Parametri che descrivono una telecamera: intrinseci, ovvero parametri che dipendono dalla telecamera e dall'ottica (distanza focale, distorsione, centro ottico, fattore scala assi); estrinseci, parametri che dipendono dalla posizione della telecamera rispetto al sistema di riferimento *world*.

Le proiezioni descritte nel capitolo precedente permettono di associare a ciascun punto p bidimensionale di coordinate $p = [u, v]^T$ (appartenente ad un piano) una sua rappresentazione 3D, chiamata coordinata omogenea $\tilde{p} = [hu, hv, h]^T$, dove h è una costante reale. Prendendo $h = 1$ si ottengono le coordinate $p = [u, v, 1]^T$ denominare vettore esteso di p. Nella stessa maniera, ogni punto 3D di coordinate cartesiane $P = [x, y, z]^T$ può essere rappresentato in coordinate omogenee da un vettore 4D $\tilde{P} = [hx, hy, hz, h]^T$. Ancora una volta si prende h reale e ponendo $h = 1$ si ottiene $P = [x, y, z, 1]^T$, ovvero il vettore esteso di $\mathbf{P}$. Le coordinate di $P = [x, y, z]^T$ si ottengono da $P = [hx, hy, hz, h]^T$ dividendo per la quarta coordinata h. La rappresentazione di p in coordinate omogenee permette di modificare le seguenti relazioni

$$u - c_x = f\frac{x}{z} v - c_y = f\frac{y}{z}$$

Per portarle alla forma matriciale

$$Z \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & c_x \\ 0 & f & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{B.1}$$

Da notare che il primo membro presenta p in coordinate omogenee bidimensionali mentre il secondo membro presenta $\mathbf{P}$ in coordinate cartesiane tridimensionali. Per ottenere $\mathbf{P}$ espresso in coordinate omogenee basterà aggiungere una colonna di zeri alla destra della matrice. I sensori digitali (CCD o CMOS) impiegati nelle telecamere sono solitamente costituiti da una matrice planare di celle fotosensibili di forma rettangolare. Ogni cella avrà una dimensione orizzontale k_U e verticale k_V, come dimostrano queste figure:

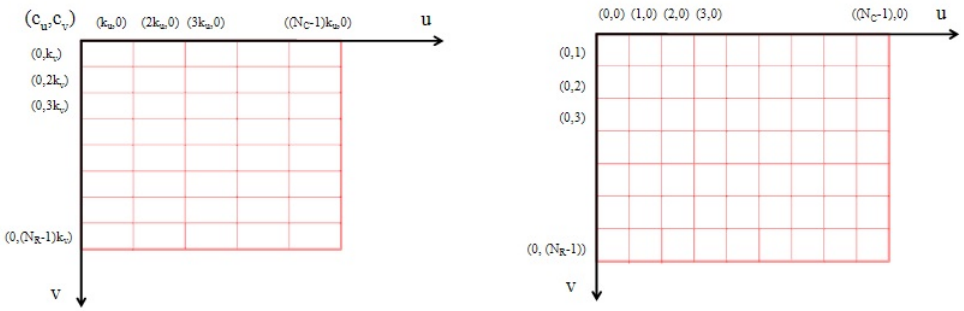


Figura B.1: a) Finestra rettangolare di un reticolo non normalizzato. b) Finestra rettangolare di un reticolo normalizzato.

Date le dimensioni del sensore, è utile per i calcoli successivi riferirci ad una finestra rettangolare normalizzata avente N_C colonne e N_R righe. In modo da lavorare con reticoli normalizzati di origine $(0, 0)$ e coordinate pixel unitarie $u_S \epsilon [0, 1, 2, ...N_C - 1]$, $vs \epsilon [0, 1, 2, ...N_R - 1]$ la relazione precedente viene rimpiazzata da:

$$z \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

dove $\mathbf{K}$ è la matrice dei parametri intrinseci definita

$$K = \begin{bmatrix} f_x & o & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

Con $f_x = fk_u$ la lunghezza focale dell'ottica nell'asse x, $f_y = fk_v$ la lunghezza focale nell'asse y, c_X e c_Y le coordinate dell'intersezione degli assi focali con il piano dei sensori. Tutte queste quantità sono espresse in $[pixel]$, f è espresso in $[mm]$ ed infine k_U e k_V sono assunti in $[pixel]/[mm]$. In alcune situazioni pratiche è conveniente rappresentare i punti 3D senza esprimerli in sistema di riferimento sensore, ma bensì al sistema convenzionale *World*. Un punto 3D avrà quindi coordinate $P_W = [x_W, y_W, z_W]^T$. La relazione che lega la rappresentazione di ogni punto riferito alle coordinate sensore con le coordinate *world* è la seguente:

$$P = RP_W + t$$

Dove R e t sono rispettivamente la matrice di rotazione e di traslazione. Rappresentando P_W in coordinate omogenee si avrà $\tilde{P}_W* = [hx_W, hy_W, hz_W, h]^T$ e, ponendo $h = 1$ la relazione può essere riscritta come:

$$P = [R, t]\tilde{P}_W$$

Quindi, la relazione tra punti di scena in coordinate omogenee e i corrispettivi in coordinate *world* diventa:

$$z \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = KP = K[R, t]\tilde{P}_W = M\tilde{P}_W = M \begin{bmatrix} x_W \\ y_W \\ z_W \\ 1 \end{bmatrix}$$

dove M sta per la matrice 3x4 di proiezione

$$M = K[R, t]$$

Ovviamente **M** dipende da parametri intrinseci (K) e da parametri estrinseci (R e t) del sistema immagine.

Nel modello reale di una telecamera si deve tener conto a distorsioni dovute alla non perfetta centratura dell'ottica, imperfezioni nella fabbricazione del sensore e distorsioni a botte o a cuscino dovute alle lenti dell'ottica. Di conseguenza le coordinate $\hat{p}$ del pixel associato al punto reale P non sono soddisfatte dalle equazioni precedenti. Le coordinate corrette del pixel (u, v) si ottengono dalle coordinate distorte (u^D, v^D) misurate attraverso la trasformazione per distorsione $\mathbf{Y}$:

$$p_T = Y^{-1}(\hat{p}_T)$$

Il modello Heikkila, ovvero il modello anti-distorsione è uno dei più utilizzati oggigiorno:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \Psi^{-1}(\hat{p}_T) = \begin{bmatrix} \hat{u}(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) + 2d_1\hat{v} + d_2(r^2 + 2\hat{u}^2) \\ \hat{v}(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) + d_1(r^2 + 2\hat{v}^2 + 2d_2\hat{u}) \end{bmatrix} \quad \text{(B.2)}$$

Dove $r = \sqrt{(\hat{u} - c_x)^2 + (\hat{v} - c_y)^2}$ e i parametri $k_i i = 1, 2, 3$ sono costanti per la distorsione radiale e $d_1 d_2$ sono costanti per la distorsione tangenziale. Questo modello quindi riassume un numero di cinque costanti, tuttavia modelli ben più complessi sono utilizzati in altre applicazioni.

Le azioni descritte in questo capitolo, ovvero l'operazione di determinazione dei parametri intrinseci ed estrinseci prende il nome di calibrazione.

Appendice C

Matrici di rototraslazione

C.0.4 Rotazioni assolute e relative

Le matrici di rotazione fondamentali rappresentano delle rotazioni del sistema di riferimento attorno agli assi X, Y e Z.

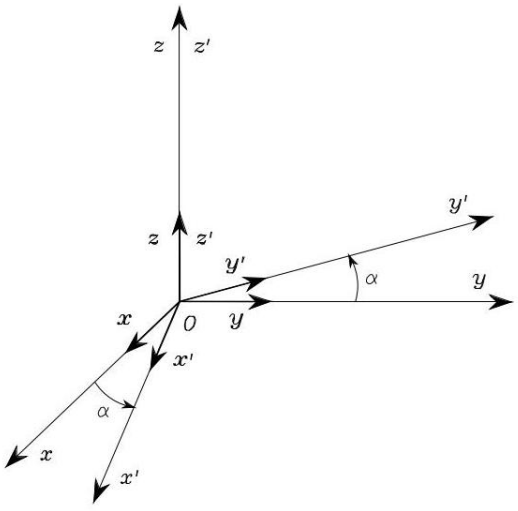


Figura C.1: Rotazione di un angolo α attorno all'asse Z.

$$R_z(\alpha) = \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 & 0 \\ \sin\alpha & \cos\alpha & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{C.1}$$

$$R_y(\beta) = \begin{bmatrix} \cos\beta & 0 & \sin\beta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\beta & 0 & \cos\beta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{C.2}$$

$$R_x(\gamma) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\gamma & -\sin\gamma & 0 \\ 0 & \sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{C.3}$$

Le matrici di rotazione fondamentali possono essere composte tra loro per costruire nuove matrici di rotazione che consentano la rappresentazione di relazioni più complesse tra sistemi di coordinate con le origini coincidenti.

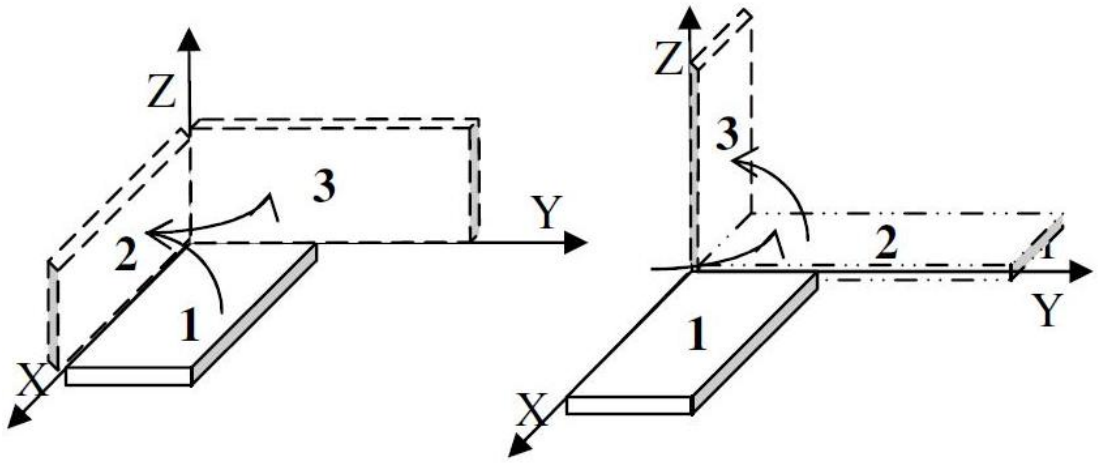


Figura C.2: Rotazioni successive attorno ad assi fissi. Si può notare che non sono commutative.

In particolare, se un sistema di coordinate è ottenuto da una rotazione del sistema di coordinate originale intorno all'asse x di un angolo θ seguita da una rotazione intorno all'asse y di un angolo γ in base alle equazioni relative alle rotazioni fondamentali, le coordinate $\mathbf{q}_1$ e $\mathbf{q}_2$ sono in relazione secondo la seguente matrice di rotazione:

$$\mathbf{q}_0 = R_2^0 \mathbf{q}_2 = R_{x,\theta} R_{y,\gamma} \mathbf{q}_2 = \begin{bmatrix} \cos\gamma & 0 & \sin\gamma \\ \sin\gamma\sin\theta & \cos\theta & -\cos\gamma\sin\theta \\ -\sin\gamma\sin\theta & \sin\theta & \cos\gamma\cos\theta \end{bmatrix} \mathbf{q}_2 \qquad \text{(C.4)}$$

La trasformazione inversa da $\mathbf{q}_0$ a $\mathbf{q}_2$ è data dalla matrice $R_0^2 = R_2^{0T}$ trasposta della matrice diretta, e quindi anche sua inversa in quanto matrice ortonormale. Questa proprietà tuttavia non è da confondere con la commutatività di due matrici di rotazione, che non sussiste (vedi figura(C.2)), ad eccezione del caso banale in cui le due rotazioni avvengono intorno allo stesso asse.
Per mostrare che le rotazioni non godono della proprietà commutativa si consideri la rotazione commutata rispetto a quella dell'equazione (C.1), cioè la rotazione di un angolo γ intorno all'asse y seguita da una rotazione di un angolo θ intorno all'asse x. La matrice relativa è data da

$$\mathbf{q}_2 = R_2^0 \mathbf{q}_0 = R_{y,\gamma} R_{x,\theta} \mathbf{q}_0 = \begin{bmatrix} \cos\gamma & \sin\gamma\sin\theta & \sin\gamma\cos\theta \\ 0 & \cos\theta & -\sin\theta \\ -\sin\gamma & \cos\gamma\sin\theta & \cos\gamma\cos\theta \end{bmatrix} \mathbf{q}_0 \qquad \text{(C.5)}$$

che evidentemente è molto diversa dalla matrice nell'equazione (C.1).

E' facile dimostrare inoltre, grazie alle proprietà delle matrici ortonormali, che per ottenere una composizione di rotazioni successive attorno ad assi fissi è necessario pre-moltiplicare le varie matrici di rotazione, mentre per ottenere la matrice di rotazione dovuta a successive rotazioni attorno ad assi mobili, ovvero relativi alla rotazione precedente, è necessario post-moltiplicare le varie matrici

di rotazione elementari.
Di conseguenza se applichiamo delle successive rotazioni R_{12}, R_{23} attorno agli assi del sistema di riferimento fisso, al vettore $\mathbf{v}_1$, otterremo $\mathbf{v}_3 = R_{23}R_{12}\mathbf{v}_1$. Sarà dunque $R_{13} = R_{23}R_{12}$ e la sua inversa $R_{31} = R_{21}R_{23}$.
Se invece le rotazioni fossero computate attorno agli assi mobili, l'ordine delle matrici sarebbe stato invertito.

C.0.5 Angoli di Eulero

La strategia più diretta per la selezione dei parametri minimi descriventi l'orientamento di una terna di riferimento consiste nel caratterizzare la matrice di rotazione in base alla composizione di tre rotazioni successive intorno a tre assi coordinati. I tre angoli associati alle rotazioni vengono denominati angoli di Eulero.
L'arbitrarietà degli angoli di Eulero consiste nel fatto che ciascuna delle tre rotazioni può essere effettuata intorno a un qualsiasi asse coordinato. Tuttavia, condizione necessaria (e sufficiente) perché i tre angoli derivanti da questa caratterizzazione siano indipendenti è che ogni coppia di rotazioni successive avvenga intorno ad assi coordinati diversi.
Vi sono dunque 27 possibili combinazioni di rotazioni, corrispondenti a rotazioni successive intorno ad assi coordinati diversi. Per ogni combinazione, la relativa terna di Eulero viene denominata terna XYZ, o terna YXY, e via di seguito.

La convenzione più usata per gli angoli de Eulero è quella associata alla terna $''ZYZ''$, caratterizzata dalle seguenti operazioni:
1. Rotazione di un angolo α intorno all'asse z;
2. Rotazione di un angolo β intorno all'asse y^I (corrente);
3. Rotazione di un angolo γ intorno all'asse z^{II} (corrente);
La convenzione degli angoli di Eulero prevede che tali rotazioni vengano via via riferite agli assi trasformati secondo l'ultima rotazione effettuata. In base a quanto illustrato nell'equazione (C.1), esse corrispondono dunque a matrici di rotazione che vanno via via a post-moltiplicare le rotazioni precedenti.

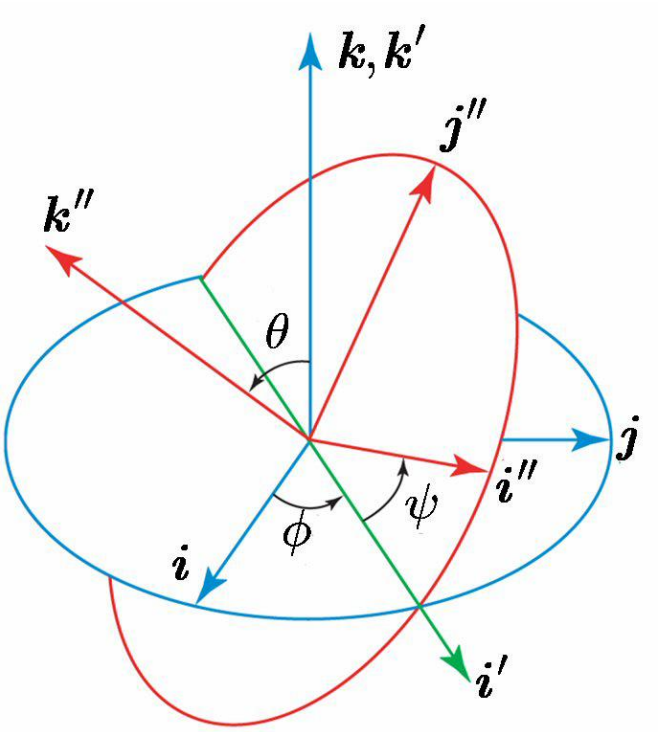


Figura C.3: Angoli di Eulero ZYZ, il sistema fisso XYZ è rappresentato in blu, e il sistema ruotato $X'Y'Z'$ è rappresentato in rosso. La linea dei nodi, indicata con N, è rappresentata in verde.

Considerando dapprima il problema della determinazione della matrice di rotazione R_{ZYZ} a partire dai valori dei tre angoli α, β e γ, si può scrivere la seguente relazione:

$$R_{ZYZ}(\alpha,\beta,\gamma) = R_{Z\alpha}R_{Y,\beta}R_{Z,\gamma} = \begin{bmatrix} C\alpha C\beta C\gamma - S\alpha S\gamma & -C\alpha C\beta S\gamma - S\alpha C\gamma & C\alpha S\beta \\ S\alpha C\beta C\gamma - C\alpha S\gamma & -S\alpha C\beta S\gamma - C\alpha C\gamma & S\alpha S\beta \\ -S\beta C\gamma & S\beta S\gamma & C\beta \end{bmatrix} \tag{C.6}$$

Data una terna α, β e γ di angoli $''ZYZ''$, l'equazione permette di ricavare la matrice di trasformazione corrispondente. Il procedimento inverso (ovvero il calcolo degli angoli corrispondenti ad un determinato orientamento, in base all'espressione della relativa matrice di rotazione R_{ZYZ}) è anche di interesse, ma corrisponde ad un problema algebrico più articolato.
Per la soluzione di quest'ultimo problema è utile introdurre la funzione $(x; y) \rightarrow$

$atan2(x;y)$ che associa ad ogni coppia di ingressi $(x;y)$ un angolo θ tale che $\sin\theta = x/(x^2+y^2)$ e $\cos\theta = y/(x^2+y^2)$.
Questa funzione viene anche denominata *arcotangente a 4 quadranti* in quanto, al contrario della classica funzione arcotangente non è soggetta all'indeterminazione tra il primo e il terzo (similarmente, il secondo e il quarto) quadrante. Inoltre, i punti di singolarità in $\pi/2 + k * \pi$ caratterizzanti la funzione classica $atan(x;y)$ non sono presenti in questo caso, grazie al fatto che $atan2(x;y)$ è funzione di due argomenti.
Procediamo ora alla determinazione della trasformazione inversa dell'equazione matriciale espressa sopra. Essendo la matrice di rotazione:

$$\begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} = \begin{bmatrix} C\alpha C\beta C\gamma - S\alpha S\gamma & -C\alpha C\beta S\gamma - S\alpha C\gamma & C\alpha S\beta \\ S\alpha C\beta C\gamma - C\alpha S\gamma & -S\alpha C\beta S\gamma - C\alpha C\gamma & S\alpha S\beta \\ -S\beta C\gamma & S\beta S\gamma & C\beta \end{bmatrix} \tag{C.7}$$

Sfruttando le uguaglianze membro a membro e sottraendo una riga all'altra, determiniamo i vari angoli, in particolare

$$\beta = atan2(\sqrt{r_{31}^2 + r_{32}^2}; r_{33}) \tag{C.8}$$

Se $S\beta$ non è $= 0$, allora

$$\gamma = atan2(r_{32}; -r_{31}) \tag{C.9}$$

$$\alpha = atan2(r_{23}; r_{13}) \tag{C.10}$$

Se invece $S\beta = 0$, allora le rotazioni α e γ e avvengono intorno allo stesso asse (eventualmente con verso opposto), quindi si può scegliere arbitrariamente $\alpha = 0$

cosicché $\sin\alpha = 0$ e $\cos\alpha = 1$, per poi determinare

$$\gamma = atan2(r_{21}; r_{11}) \tag{C.11}$$

C.0.6 Angoli di Cardano

Come gli angoli di Eulero, gli angoli di Cardano (o di Tait-Bryan) rappresentano una successione di rotazioni, ma attorno agli assi coordinati del sistema fisso, non attorno a quelli del sistema solidale.

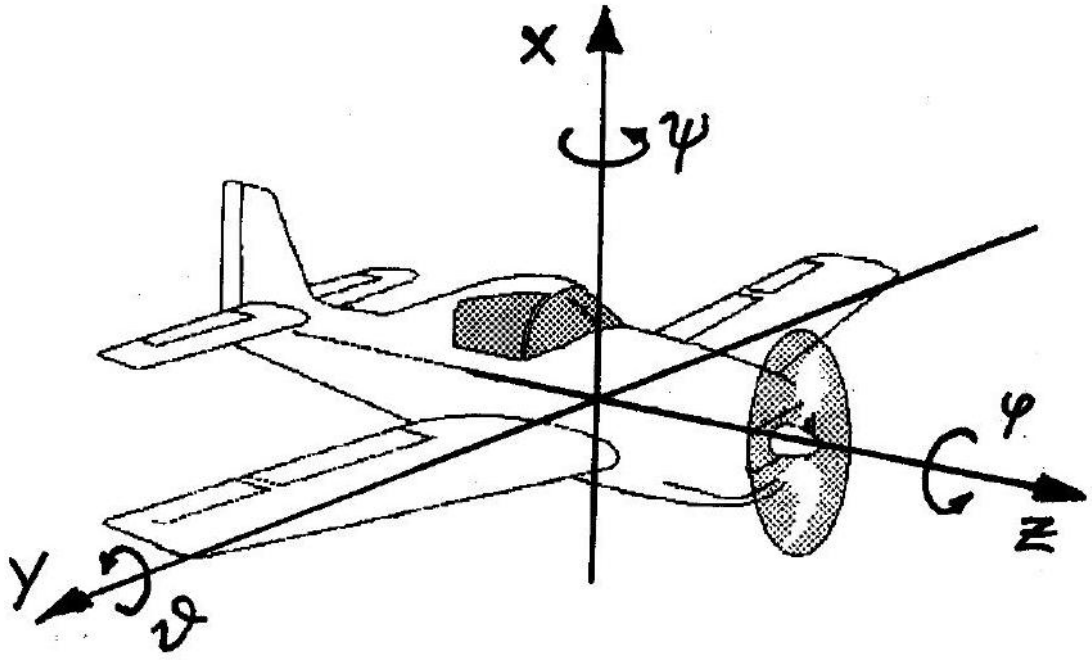


Figura C.4: Rappresentazione degli angoli di RPY

Una convenzione per la rappresentazione minima dell'orientamento particolarmente adottata in campo aeronautico è la convenzione RPY, dove R sta per rollio (*roll*), P sta per beccheggio (*pitch*) e Y sta per imbardata (*yaw*). Questa

convenzione é bene interpretabile facendo riferimento all'assetto di un aeroplano sul quale sia stato fissato un sistema di riferimento il cui asse z è disposto lungo la carlinga, il cui asse y è disposto nella direzione dell'apertura alare e il cui asse x è disposto di conseguenza. Secondo la convenzione RPY, gli angoli di rollio ρ, di beccheggio θ e di imbardata ϕ vengono definiti eseguendo tre rotazioni successive, tutte intorno agli assi del sistema di riferimento originale, secondo la sequenza:
1. Rotazione di un angolo intorno all'asse x;
2. Rotazione di un angolo intorno all'asse y (originale);
3. Rotazione di un angolo intorno all'asse z (originale).
In base a quanto illustrato nel paragrafi precedenti, le tre rotazioni sopra elencate corrispondono a matrici di rotazione che vanno via via a pre-moltiplicare le rotazioni precedenti, infatti eseguendo le rotazioni successive rispetto al sistema di riferimento originale, l'effetto è quello che si otterrebbe se la rotazione in oggetto fosse anteposta a quelle già effettuate.
Pre-moltiplicando, dunque, le matrici di rotazione relative alle trasformazioni sopra elencate, si può procedere alla determinazione della matrice di rotazione $R_{RPY}(\rho;\phi;\theta)$ analoga a quella riportata in equazione C.6 per il caso degli angoli ZYZ:

$$R_{RPY}(\rho,\theta,\phi) = R_{Z\rho}R_{Y,\theta}R_{X,\phi} = \begin{bmatrix} C\rho C\theta & C\rho S\theta S\phi - S\rho C\phi & C\rho C\theta C\phi + S\rho S\phi \\ S\rho C\theta & S\rho S\theta S\phi - S\rho S\phi & C\rho C\theta C\phi + C\rho S\phi \\ -S\theta & C\theta S\phi & C\theta C\phi \end{bmatrix} \tag{C.12}$$

Si osservi che gli angoli di RPY corrispondono ad una delle 27 possibili scelte per gli angoli di Eulero indicate nel paragrafo precedente. In particolare, essi corrispondono agli angoli di Eulero ZYX. La determinazione della trasformazione inversa alla C.12 può essere eseguita in maniera parallela a quanto fatto nel paragrafo precedente per il caso degli angoli ZYZ.

C.0.7 Rotazione attorno ad un asse

Per questo lavoro di tesi è apparso necessario studiare la composizione di rotazioni che porta un oggetto ed il suo sistema di riferimento, a ruotare attorno ad

un asse qualsiasi nello spazio.

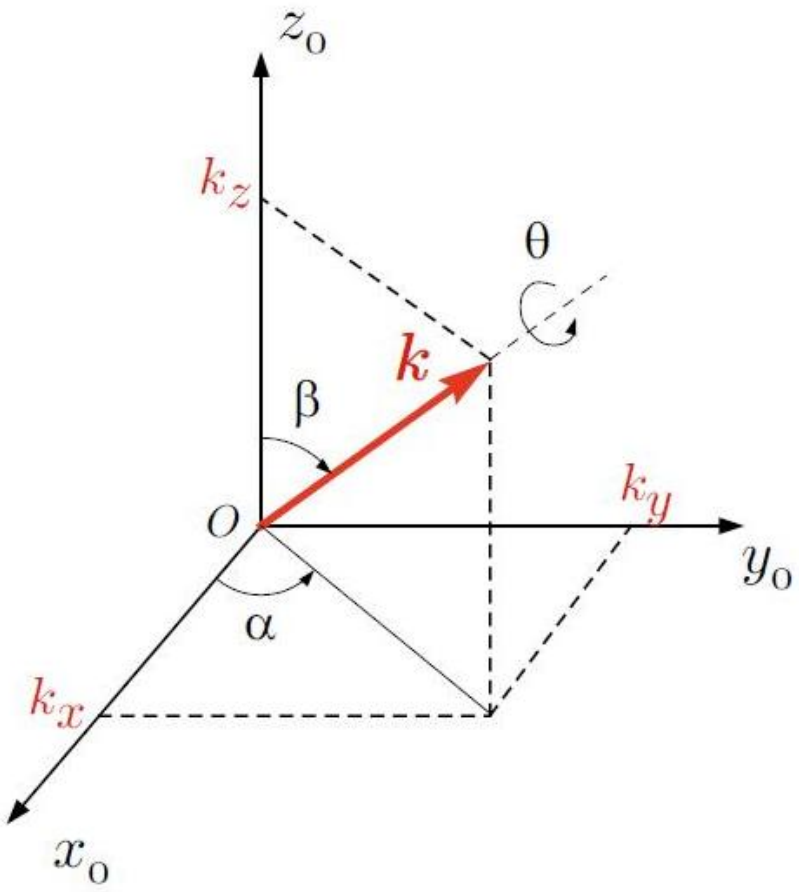


Figura C.5: Rotazione di un angolo θ attorno ad un asse generico.

Una qualsiasi rotazione di un corpo rigido è esprimibile tramite una matrice di rotazione e quindi abbiamo trovato la matrice che esprime la rotazione attorno ad un asse generico.
Con riferimento alla figura, si consideri il caso in cui rispetto alla terna di riferimento $0XYZ$, l'asse di rotazione abbia coordinate $(r_x; r_y; r_z)$, con $(r_x^2 + r_y^2 + r_z^2) = 1$. In altre parole, si supponga che il vettore r mostrato in figura C.5 sia di fatto un versore. In questo caso, la rotazione di un angolo θ intorno all'asse individuato da r può essere descritta dalla composizione di rotazioni elementari come segue:
1. rotazione di un angolo $-\alpha$ intorno all'asse z per portare l'asse r sul piano verticale individuato dagli assi gli assi x e z;
2. rotazione di un angolo $-\beta$ intorno all'asse y per sovrapporre l'asse r all'asse z;

3. rotazione di un angolo θ intorno all'asse $z = r$;

4. rotazione di un angolo β intorno all'asse y;

5. rotazione di un angolo α intorno all'asse z.

Si noti che le ultime due rotazioni vengono eseguite per riportare l'asse r nella posizione originaria. In particolare, poiché la rotazione da rappresentare avviene intorno all'asse r, tutti i punti dello spazio su questo asse non devono subire nessuno spostamento.

La procedura sopra elencata può essere descritta in termini di matrici di rotazione fondamentali tramite la formula seguente (si noti che tutte le rotazioni sopraelencate sono riferite alla terna fissa dunque corrispondono a matrici di rotazione che vengono via via pre-moltiplicate):

$$R_{r\theta} = R_{Z\alpha} R_{Y\beta} R_{Z\theta} R_{Y-\beta} R_{Z-\alpha} \tag{C.13}$$

nella quale è utile eliminare la dipendenza da α e β, esprimendola in funzione delle componenti di r rispetto al sistema di riferimento $0XYZ$). In particolare, poiché r ha norma unitaria, le seguenti relazioni derivano da semplici argomentazioni geometriche:

$$r_x = \sqrt{r_x^2 + r_y^2} \cos\alpha \tag{C.14}$$

$$r_y = \sqrt{r_x^2 + r_y^2} \sin\alpha \tag{C.15}$$

$$\sqrt{r_x^2 + r_y^2} = \sin\beta \tag{C.16}$$

$$r_z = \cos\beta \tag{C.17}$$

Da queste relazioni si ricava sostituendo seni e coseni di α e β e moltiplicando:

$$R_{r\theta} = \begin{bmatrix} r_x^2(1-C_\theta)+C_\theta & r_x r_y(1-C_\theta)-r_z S_\theta & r_x r_z(1-C_\theta)+r_y S_\theta \\ r_x r_y(1-C_\theta)+r_z S_\theta & r_y^2(1-C_\theta)+C_\theta & r_y r_z(1-C_\theta)-r_x S_\theta \\ r_x r_z(1-C_\theta)-r_y S_\theta & r_y r_z(1-C_\theta)+r_x S_\theta & r_z^2(1-C_\theta)+C_\theta \end{bmatrix} \tag{C.18}$$

che rappresenta la rotazione di un angolo θ della terna originaria intorno all'asse arbitrario r.

Appendice D

Glossario

In questa sezione sono presenti il glossario dei termini usati e un elenco di chiavi di ricerca (*keyword*), con relativa traduzione italiana, utili per il reperimento degli argomenti sui motori di ricerca internazionali.

D.1 Glossario di riferimento

- **ADEPT DESKTOP**
  E' un ambiente grafico per il test e la programmazione dei sistemi robot *Adept*. La programmazione viene eseguita in linguaggio $V+$.

- **CINEMATICA**
  Ramo della meccanica che si occupa di descrivere quantitativamente il moto dei corpi senza riferimento alle cause del moto.

- **CONTROLLER**
  Con *controller*, o *controllore*, si indica un sistema (del tipo elettronico) il cui compito è far svolgere ad una macchina un'azione determinata da un segnale di ingresso. Infatti l'obiettivo del controllore, nell'esercizio dell'azione di controllo, è quello di far sì che l'andamento della variabile controllata non si discosti troppo dall'andamento del segnale di riferimento stesso.

- **ENCODER**
  Dispositivo che permette di misurare una rotazione attraverso un lettore

ottico ed un apposito disco forato. Il valore fornito in ingresso viene espresso in codifica di 2^n bit.

- **END-EFFECTOR**

  Con questo termine viene indicato l'ultimo membro della catena cinematica annessa al robot. Più precisamente si indica l'attrezzo che permette il contatto del robot con altri oggetti. Solitamente esso è costituito da una pinza, ma nel varieggiato mondo industriale può essere costituito da sensore anti-collisione, forbici, mandrino, saldatore, spruzzatori ecc...

- **EURISTICA**

  La parola *euristica* indica la parte della ricerca il cui compito è quello di favorire l'accesso a nuovi sviluppi teorici o a scoperte empiriche. Si definisce, infatti, procedimento euristico, un metodo di approccio alla soluzione dei problemi che non segue un chiaro percorso, ma che si affida all'intuito e allo stato temporaneo delle circostanze, al fine di generare nuova conoscenza.

- **GIUNTO PRISMATICO**

  Un giunto è un dispositivo capace di rendere solidali tra loro due estremità d'albero in modo tale che l'uno possa trasmettere un momento torcente all'altro; con prismatico si intende un giunto che realizza un moto relativo di traslazione tra due bracci.

- **GIUNTO ROTAZIONALE**

  Un giunto è un dispositivo capace di rendere solidali tra loro due estremità d'albero in modo tale che l'uno possa trasmettere un momento torcente all'altro; con rotazionale si intende che tra il primo componente ed il secondo si immette un solo grado di libertà e si instaura una rotazione.

- **GRADI DI LIBERTA'**

  Per gradi di libertà di un corpo si intende il numero di coordinate generalizzate necessarie per descrivere il suo moto.

- **IMAGING**

  Processo durante il quale si immagazzinano informazione sotto forma di immagini.

- **INTEGER**

  In informatica si definisce intero (o nella sua forma inglese integer, spesso abbreviato in int) ogni tipo di dato che possa rappresentare un sottoinsieme dell'insieme matematico dei numeri interi.

- **LASER**

  Dispositivo in grado di emettere luce monocromatica e concentrata in un sol punto o in uno fascio.

- **MATCHING**

  Operazione di accoppiamento tra due nuvole di punti o tra due superfici.

- **MATLAB**

  è un ambiente per il calcolo numerico e l'analisi statistica fondato negli anni '70. Si basa su un linguaggio di programmazione definito $MathWorks$ ed è disponibile per tutti i sistemi operativi. Il suo campo d'applicazione è molto vasto in quanto è in continua crescita e sviluppo.

- **METODO DELLA MASSIMA VEROSIMIGLIANZA**

  è un procedimento matematico che consiste nel massimizzare la funzione di verosimiglianza, definita in base alla probabilità di osservare una data realizzazione campionaria.

- **PENDANT**

  Con *Pendant*, o *Robot-Pendant* si intende un dispositivo portatile che permette di comandare il robot tramite l'ausilio di comandi e, nelle versioni moderne, di interfacce grafiche.

- **PIXEL**

  In computer grafica, con il termine pixel si indica ciascuno degli elementi puntiformi che compongono la rappresentazione di una immagine raster digitale, ad esempio su un dispositivo di visualizzazione o nella memoria di un computer.

- **REVERSE ENGINEERING**

  Analisi dettagliata del funzionamento o della forma di un oggetto. Questo

procedimento viene spesso iniziato con la scansione laser dell'oggetto e la ricostruzione tramite software CAD 3D.

- **TASTATORE**

  Strumento utilizzato in robotica e nella *reverse engineering* per fornire un contatto con la superficie o l'oggetto.

- **TRASFORMATA DI HOUGH**

  Tecnica per l'elaborazione digitale delle immagini che permette il riconoscimento di forme geometriche arbitrariamente definite.

Bibliografia

[1] K. Khoshelham. Accuracy analysis of Kinect depth data. ITC Faculty, University of Twente. International Archieves of Photogrammetry, Remote and Spatial Information Scienses, Calgary, Canada, August 2011.

[2] M.R. Andersen, T. Jensen, P. Lisouski, A.K. Mortensen, M.K. Hansen, T. Gregersen and P. Ahrendt. Kinect Depth Sensor Evaluation for Computer Vision Applications. 2012.

[3] Mikkel Viager. Analysis of Kinect for mobile robots. 2011.

[4] C. Dal Mutto, P. Zanuttigh, G.M. Cortellazzo. Time-of-Flight Cameras and Microsoft Kinect. 2013.

[5] Stefano Tonello. Bin Picking Robot: Algoritmi di visione e framework software. Università degli studi di Padova. Master Thesis, 2008, Padova, Italia.

[6] A. Rossi. Appunti delle lezioni, DIMEG Robotics, Università degli studi di Padova, Italia, 2013.

[7] Mark Theodore Draelos, Edward Grant. The Kinect up close: Modifications for short-range depth imaging. Master thesis 2012, Raleigh, North Carolina, USA.

[8] Jan Smisek, Michal Jancosek and Tomas Pajdla. CMP, Departement of Cybernetics, FEE, Czech Technical University in Prague, 2012.

[9] Matthias Nieuwenhiusen, David Droeschel, Dirk Holz, Jorg Stuckler, Alexander Berner, Jun Li, Reinhard Klein and Steve Behnke. Mobile Bin Pic-

king an Anthropomorphic Service Robot. IEEE International Conference on Robotics and Automation (ICRA), Karlsruhe, Germany, May 2013.

[10] Paul J. Besl and Neil D. McKay. A Method for Registration of 3-D Shapes. IEEE Transactions on pattern and machine intelligence, vol 14, no.2, February 1992.

[11] Thomas Jimesh. Empirical Evaluation of a Machine Vision System for Random Bin Picking Application. University of Applied Sciences Bonn-Rhein-Sieg. Master Thesis 2008, Boston, MA, USA.

[12] Martin Kvalbein. The use of a 3D sensor for robot motion compensation. University of Applied Sciences Bonn-Rhein-Sieg. Master Thesis 2008, Boston, MA, USA.

[13] K. Khoshelham. Accuracy analysis of Kinect depth data. 2013.

[14] Thomas Jimesh. Empirical Evaluation of a Machine Vision System for Random Bin Picking Application. University of Applied Sciences Bonn-Rhein-Sieg. Master Thesis 2008, Boston, MA, USA.

[15] Zhengyou Zhang. A Flexible New Technique for Camera Calibration. IEEE Transactions on Pattern Analysis and Machine Intelligence. Volume 22. Year 2000. Pages 1330-1334.

[16] Sergi Foix, Guillem Aleny, Carme Torras. Lock-in Time-of-Flight (ToF) Cameras: A Survey. IEEE Sensors Journal. Volume 11. Year 2011. Pages 1917-1926.

[17] Ayako Takenouchi, Naoyoshi Kanamaru, Makoto Mizukawa. Hough-space-based Object Recognition Tightly COupled with Path Planning for Robust and Fast Bin-picking. Proceeding of the 1998 IEEE/RSJ Intl. Conference on Intelligent Robots and Systems, Victoria, Canada.

[18] Hans Martin and Jacob Wilm. Evaluation of surface registration algorithms for PET motion correction. Batchelor Thesis, Technical University of Denmark, 2010.

[19] *http://www.adept.com*

[20] *http://www.primesense.com*

[21] *http://www.fotonic.com*

[22] *http://www.mesa-imaging.ch*

[23] *http://www.geomagic.com*

Ringraziamenti

Ringrazio la mia famiglia per avermi dato la possibilita' di intraprendere gli studi universitari e tutti quelli che mi hanno supportato. Un ringraziamento va' al professor Giulio Rosati e all'ing. Simone Minto per gli insegnamenti datami. Si ringrazia inoltre il laboratorio di Metrologia Geometrica ed Industriale del **DIMEG** per la scansione effettuata.