

Università degli Studi di Padova

Facoltà di Ingegneria

Corso di Laurea Magistrale in Ingegneria Meccatronica

tesi di laurea

Sistemi di visione 3D evoluti per Bin Picking di oggetti con geometrie complesse

Relatore: Ch.mo Prof. Giovanni Boschetti

Laureando: Dario De Rossi

A.A. 2022/2023

Indice

Introduzione	1
1 Il Bin Picking industriale	3
1.1 Introduzione	3
1.2 Sistemi di Visione 3D	6
1.2.1 Triangolazione Laser	6
1.2.2 Tempo di Traiettoria	8
1.2.3 Luce Strutturata	14
1.2.4 Visione Stereoscopica	21
2 Cella robotizzata in laboratorio	25
2.1 Introduzione	25
2.2 Robot	25
2.3 Sistema di visione	26
2.4 Fanuc Bin Picking	30
2.4.1 Parts List Manager	31
2.4.2 Interference Avoidance	32
2.4.3 Programmazione: Vision e Teaching Pendant	36
3 Riconoscimento di oggetti con geometrie complesse	47
3.1 Introduzione	47
3.2 Test sistema di visione e Vision Tools	48
3.2.1 GPM Locator Tool	49
3.2.2 3D One-Sight-Model Locator Tool	57
4 Kitting motore endotermico	69
4.1 Introduzione	69
4.2 Vision Process setting	74
4.3 Interference Avoidance setting	78
4.4 Parts List Manager setting	80
4.5 TP Programming	82
Conclusioni	91

Introduzione

Il presente documento nasce con lo scopo di illustrare le tecnologie ad oggi disponibili per sistemi di visione tridimensionali, approfondendo maggiormente quelle impiegate nell'industria, in particolar modo nell'applicazione industriale del bin picking. Nello specifico, questa attività viene descritta nel Cap.1, dove vengono illustrate le principali criticità, ed i motivi per i quali è necessario l'impiego di sistemi di visione 3D. Questi sono sempre più diffusi, trovano infatti numerose applicazioni, dall'automotive all'aerospace, nei dispositivi portatili come gli smartphone e nell'industria, nella quale la loro introduzione ha permesso l'automatizzazione di svariati processi industriali. Nel medesimo capitolo vengono descritte le tecnologie utilizzate in tali sistemi, includendo esempi di sensori reali per applicazioni industriali e non. L'attività di tesi è stata di tipo sperimentale, si è svolta nel laboratorio di robotica presso il Dipartimento di Tecnica e Gestione dei Sistemi Industriali (DTG), dove è presente una cella robotizzata, descritta nel Cap.2, nel quale viene trattata nel dettaglio anche la logica di programmazione per il riconoscimento 3D di oggetti e per il controllo del robot. Di tale cella è stato approfonditamente testato il sistema di visione (Cap.3), suo componente principe nell'ottica di questa tesi, con l'obiettivo di valutarne e verificarne le performance, testandolo per il riconoscimento di componenti più complessi ed in generale che, per le loro caratteristiche, rappresentano una sfida per tale sistema. Dopo aver testato tale sistema di visione, il lavoro di tesi si è concentrato sull'individuazione di un processo industriale reale, che potesse rappresentare una sfida non limitatamente alla visione, ma per la cella robotizzata nel suo complesso. Nel sistema analizzato, viene previsto di default l'utilizzo di un contenitore, nel quale sono presenti numerosi componenti, tutti dello stesso tipo, da movimentare per poi eseguire il place in una singola posizione. Al fine di andare oltre tali specifiche, l'applicazione industriale scelta, implementata in laboratorio e descritta in questa tesi è un kitting per motore endotermico (Cap.4): questa prevede diversi contenitori e differenti componenti, da movimentare e piazzare ognuno all'interno del proprio specifico slot, in una maschera appositamente preformata. Nel medesimo capitolo è descritta la programmazione specifica per tale applicazione, sia per la visione che per il robot, e vengono riportate, a corredo, immagini che illustrano i risultati per il riconoscimento degli oggetti.

Capitolo 1

Il Bin Picking industriale

1.1 Introduzione

In molti processi industriali, come l'assemblaggio o il kitting, i componenti necessari si trovano all'interno di contenitori, ad esempio scatole, e in essi sono disposti alla rinfusa, solitamente sovrapposti tra loro. Questi devono essere prelevati da tali contenitori e per farlo si hanno a disposizione principalmente due opzioni:

- *svuotamento manuale*: l'operatore provvede a prendere i componenti all'interno dei contenitori;
- *svuotamento automatizzato*: è richiesto l'utilizzo di uno o più robot, che secondo la logica di programmazione, eseguono la presa dei componenti dai contenitori.

In entrambi i casi si parla di *Bin Picking*, letteralmente *presa da cassone*, tuttavia con tale definizione ci si riferisce solitamente a un processo di presa di componenti da contenitore in modo automatizzato, quindi facendo riferimento alla seconda modalità sopra citata. Nello specifico, questa attività consiste nell'individuazione degli oggetti disposti alla rinfusa all'interno di un contenitore. L'oggetto così individuato potrà quindi essere afferrato da un robot manipolatore, per poi essere soggetto al processo industriale designato. A differenza del bin picking manuale, che vanta la massima flessibilità, visto che gestita dall'operatore, e nessun bisogno di particolari strumenti o attrezzature, il bin picking automatizzato risulta assai più complesso, in quanto deve far fronte a svariate difficoltà. I robot sono infatti privi di sensori utili per queste applicazioni, vengono per questo anche definiti *ciechi*, di conseguenza solitamente sono inseriti in contesti nei quali è richiesto eseguano sempre la stessa sequenza di movimenti. Di default quindi i robot non hanno la percezione dell'ambiente che li circonda, tantomeno degli oggetti di cui devono fare la presa. L'automatizzazione dei sistemi robot per bin picking è composta da:

- **riconoscimento componenti:** essendo come detto gli oggetti disposti alla rinfusa e probabilmente sovrapposti, sono necessari algoritmi avanzati per il riconoscimento degli stessi. Inoltre è fondamentale che il sistema sia in grado di valutare se un oggetto risulta totalmente visibile quindi raggiungibile, o se questo è incastrato o coperto quindi non raggiungibile;
- **localizzazione dei componenti:** è necessario stimare con precisione la posizione e l'orientazione nello spazio dei componenti da afferrare;
- **pianificazione del movimento del robot:** questa deve evitare che il robot comprensivo del suo organo terminale, nei movimenti, collida con
 - altri componenti nel contenitore;
 - il contenitore stesso;
 - oggetti e ambiente circostanti al manipolatore.

Questo viene definito *collision detection & avoidance* o *interference detection & avoidance*.

- **pianificazione della presa dell'oggetto:** come anticipato, è fondamentale determinare se un componente è raggiungibile, quindi del quale è possibile eseguire la presa. Questo dipende da come è disposto il pezzo rispetto agli altri nel contenitore (se coperto da altri, anche solo parzialmente, le griffe dell'organo terminale potrebbero non riuscire a passare per afferrarlo) e rispetto al contenitore stesso (se troppo vicino alle pareti ad esempio la presa non riesce perché interviene l'algoritmo di collision avoidance);

Al fine di adempiere ai task appena descritti, i robot devono essere equipaggiati con la sensoristica adatta, che, in generale può essere racchiusa nella definizione di *Sistema di Visione*, ed è necessario che questo sia tridimensionale. Nel caso di un sistema di visione 2D, infatti, l'immagine dell'oggetto target acquisita è effettivamente piatta, in quanto bidimensionale. Essa di conseguenza non fornisce nessuna informazione relativa all'altezza: ci sono coordinate per gli assi X e Y ma non per Z, indicato come asse di profondità. Il risultato è il contorno di un oggetto 3D visto da uno specifico punto di osservazione. Diversi punti di osservazione e diversi oggetti creano contorni completamente diversi, rendendo molto limitato l'utilizzo di un sistema 2D per le applicazioni dove le informazioni sulla forma sono fondamentali per lo svolgimento di un compito. Tuttavia i sistemi di visione 2D vengono ampiamente utilizzati nel settore industriale per molte mansioni come:

- verifica di caratteristiche e posizione;
- controllo delle dimensioni;
- lettura del codice a barre;

- riconoscimento caratteri;
- verifica dell'etichetta;
- controllo qualità;
- rilevatore di presenza.

In tutti questi casi l'immagine 2D viene ottenuta attraverso la variazione dei riflessi di luce lungo la superficie degli oggetti. Il contrasto dell'oggetto identificato, sia in scala di grigi che a colori, rappresenta la prima sfida per il sistema di visione 2D.

In aggiunta al precedentemente citato problema relativo alla mancata rilevazione di profondità, questi sistemi soffrono inoltre di

- Sensibilità alla luce: l'immagine dell'oggetto target è ottenuta a partire dai riflessi di luce su di esso incidente, di conseguenza variazioni nella luminosità, dovute a cambiamenti delle condizioni ambientali o a luci artificiali, possono avere un impatto negativo sul livello di accuratezza. Troppa luce, poca luce o oscuramenti nell'ambiente di lavoro possono influire negativamente sulla chiarezza dei bordi e dei lineamenti che appaiono nell'immagine 2D acquisita;
- Assenza di contrasto: considerato che il sistema di visione 2D dipende direttamente dai contrasti definiti sulla superficie dell'oggetto, risulta complesso gestire oggetti molto chiari o molto scuri.

Per i motivi appena citati, i sistemi di visione più adatti al bin picking sono di tipo tridimensionale, nel seguente Cap. [1.2](#) vengono descritte le tecnologie disponibili ed adatte ad essere impiegate in tali sistemi di visione.

1.2 Sistemi di Visione 3D

Come anticipato nel Cap [1.1](#) per il bin picking è fondamentale che al robot sia affiancato un complesso di sensori in grado di rilevare precise coordinate nello spazio. Questa caratteristica è peculiare dei *Sistemi di Visione 3D*, grazie ai quali la posizione di ogni pixel nello spazio è conosciuta.

Esistono principalmente quattro tecniche per realizzare un sistema di visione 3D, che possono essere categorizzate in sistemi ottici attivi e sistemi ottici passivi:

- Sistemi Ottici Attivi: costituiti da un emettitore che illumina un punto, una sezione o un oggetto, ed un sensore che riceve parte della luce riflessa dalla superficie illuminata. Ne fanno parte:
 - Triangolazione Laser;
 - Tempo di Traiettoria;
 - Luce Strutturata;
- Sistemi Ottici Passivi: non fanno uso di alcuna fonte di illuminazione specifica ma sfruttano la luce proveniente dall'ambiente. Appartiene a questa categoria la:
 - Visione Stereoscopica.

1.2.1 Triangolazione Laser

In questi sistemi, la fonte di illuminazione è un fascio di luce laser, da proiettare sull'oggetto di interesse, la componente riflessa viene rilevata da un sensore tipicamente CCD/CMOS. Questo metodo è chiamato triangolazione perché il punto (o linea) laser, il sensore e l'emittente laser formano un triangolo come illustrato in Fig [1.1](#)

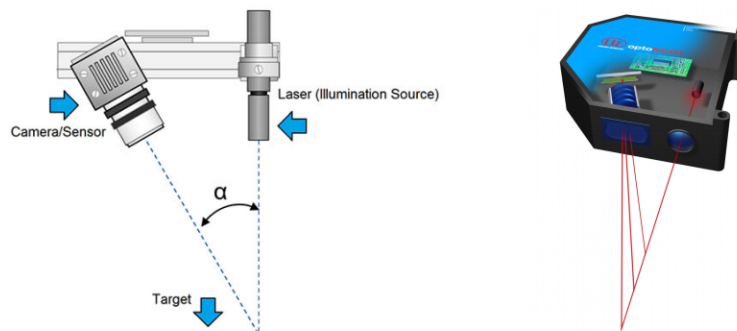


Figura 1.1: A sinistra, schema che illustra la geometria dei tre elementi fondamentali di un sistema a triangolazione laser. A destra, la disposizione di tali componenti all'interno di un sistema reale, un *optoNCDT* di *Micro-Epsilon*.

I due elementi appena citati hanno una disposizione relativa conosciuta, in questo modo è possibile triangolare la posizione di un punto della superficie nel sistema di riferimento adottato. Un esempio di geometria laser-sensore è illustrata in Fig. 1.2, in quest'ultima l'origine del sistema di riferimento è collocata nel laser, il quale proietta un fascio orientato di un angolo θ rispetto all'asse Z e dista dalla superficie stessa di una quota z . Il sensore, rispetto al laser, è posizionato ad una distanza d , detta *baseline*, e l'immagine del raggio laser riflesso dalla superficie dell'oggetto si forma ad una distanza p dal punto principale.

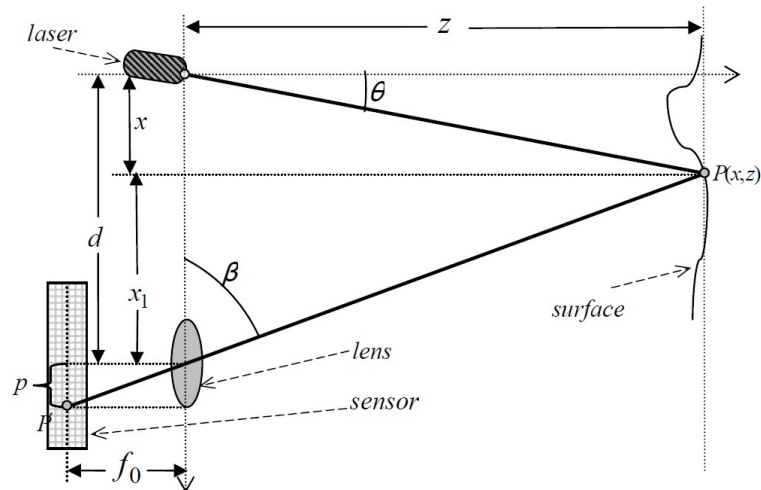


Figura 1.2: Esempio di geometria laser-sensore.

I principali vantaggi di questa tecnologia sono i seguenti:

- la luce proiettata è coerente¹, monocromatica e ad alta luminosità. Per questo è possibile utilizzare filtri ottici passa-banda atti ad isolare la luce del laser in ambienti nei quali sono presenti altre fonti di illuminazione, che recherebbero disturbo all'acquisizione;
- il raggio può essere orientato in maniera controllata mediante un sistema di specchi rotanti, in questo modo è possibile scansionare una superficie.

Gli svantaggi invece includono:

- potenziali rischi all'occhio umano dovuti all'intensità della luce prodotta dal laser;
- l'accuratezza dipende fortemente dal tipo di materiale analizzato: questa tecnologia è indicata per materiali opachi, sono invece sconsigliati materiali trasparenti o traslucidi.

¹In ottica si chiama coerenza (o coerenza di fase) la proprietà di un'onda elettromagnetica di mantenere una certa relazione di fase con se stessa durante la sua propagazione.

I dispositivi a tecnologia laser applicati al bin picking vengono definiti *3D area-scan displacement sensors*. In generale, i profilometri laser generano delle sezioni definite *slices* del componente per poi andarle ad affiancare ottenendo la mappa tridimensionale continua. Si ha la necessità di un encoder che segue l'oggetto analizzato. Gli scanner di area menzionati in precedenza fanno invece uso dei cosiddetti *specchi MEMS*, ossia dei sistemi di specchi miniaturizzati, che offrono la possibilità di variare la direzione di un fascio di luce, in questo caso della sorgente laser. In questo modo l'oggetto può rimanere fermo, è la luce che lo scannerizza. Con questa tecnologia è possibile ottenere accuratissime, dell'ordine dei μm con tempi di acquisizione intorno ai 2s.

1.2.2 Tempo di Traiettoria

Al contrario dei dispositivi a triangolazione laser, i dispositivi a tempo di traiettoria hanno illuminatore e sensore posti nella medesima posizione. Si indicano con l'acronimo *ToF (Time of Flight)*, e come indica la loro denominazione, fanno uso del tempo di volo t . Il principio di base di questi sistemi sfrutta la conoscenza della velocità della luce c e la misura di t tempo dall'invio alla ricezione della luce grazie alla riflessione sull'oggetto analizzato. Per mezzo di questi è possibile ottenere la misura della distanza dal punto di riflessione come $\frac{c \cdot t}{2}$. In questi dispositivi la scena non viene analizzata una linea per volta, ma viene effettuata una singola cattura, la scena è quindi acquisita in modo completo come per una foto. I sensori ToF utilizzano quindi i fotoni, le particelle elementari che compongono la luce, tuttavia sfruttano solamente quelli aventi lunghezza d'onda λ vicina allo spettro visibile. Nello specifico, la luce proiettata è modulata e viene misurata la differenza di fase tra questa e quella rilevata dal sensore ossia quella di ritorno dall'oggetto come riflessa. La versione più semplice di sensori ToF fa infatti uso di uno o più impulsi di luce, generati accendendo l'elemento illuminatore per un brevissimo periodo. Questi, come detto, viaggiano fino ad arrivare alla scena in esame, su essa rimbalzano, la lente direziona questa componente riflessa verso l'elemento sensibile (sensore o *focal plane array*²). Tale componente riflessa risulta essere in ritardo rispetto alla componente da cui è originata, questo valore è solitamente indicato con t_D e, nota la velocità della luce $c=300\,000\,000\text{ m/s}$ e la distanza D tra fonte di luce e scena, si può ottenere come

$$t_D = 2 \cdot \frac{D}{c} \quad (1.1)$$

Il singolo pixel presente all'interno dell'elemento sensibile è un elemento fotosensibile, ad esempio un fotodiode, questo converte la luce rilevata in una corrente elettrica. In alcuni dispositivi vengono usati gli *analog timing imagers*, nei quali i fotodiode sono connessi degli interruttori che portano la corrente appena citata a degli elementi di memoria, solitamente condensatori, che costituiscono

²sensore di immagine costituito da un vettore di pixel sensibili alla luce.

una sommatoria. Un'altra possibilità è rappresentata dai *digital timing imagers*, che fanno uso di un contatore di tempo digitale estremamente veloce, collegato ad ogni elemento fotosensibile e smette di contare quando questo è investito dalla luce. Un esempio della soluzione analogica appena descritta è illustrata in Fig. 1.3. Gli elementi G sono gli switch, controllati da un segnale avente la stessa lunghezza dell'illuminazione generata, ma quello di G2 ha un ritardo della durata dell'impulso stesso t_0 . Gli elementi S sono invece le memorie. Come si vede dal secondo diagramma, il tempo di ritardo t_D dovuto alla riflessione fa sì che solo una parte dell'impulso riflesso venga catturata in S1 passando per G1

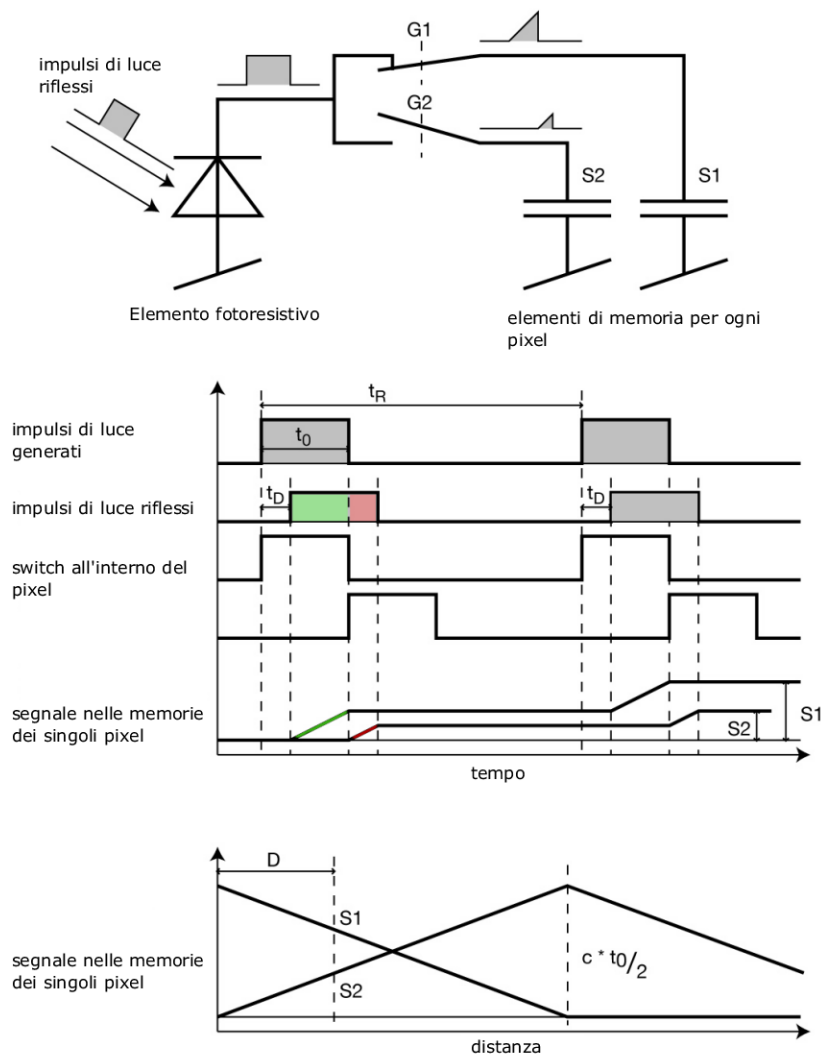


Figura 1.3: Diagrammi che illustrano il funzionamento di un sensore ToF con *analog time imagers*.

(in Fig. 1.3 quella evidenziata in verde), la restante viene immagazzinata in S2.

Dopo tale esposizione, avendo ottenuto i valori di S1 ed S2 mediante circuiteria successiva, è possibile determinare la distanza D alla quale si trova l'oggetto sul quale la luce è stata riflessa, come

$$D = \frac{1}{2} \cdot c \cdot t_0 \cdot \frac{S2}{S1 + S2} \quad (1.2)$$

Nella maggior parte dei casi però, la scena analizzata presenta fonti di luce esterne, che vengono rilevate dal sensore ToF e rappresentano una quota aggiuntiva nel segnale, sono di fatto un disturbo per il dispositivo in esame. Al fine di risolvere questo problema, nella maggioranza dei dispositivi a tempo di volo la sorgente di luce non produce solo un impulso ma una molteplicità di impulsi oppure viene modulata con una sinusoide.

I sistemi di visione a tempo di volo presentano i seguenti vantaggi:

- efficienza degli algoritmi: il processo che porta ad ottenere l'informazione sulla distanza necessita di poca potenza di calcolo;
- semplicità: essendo la lente ed il proiettore necessariamente vicini, l'insieme di componenti permette di essere montato in modo semplice all'interno di case estremamente compatti. Non sono inoltre richieste parti mobili;
- velocità: come detto in precedenza, un'acquisizione di un sistema ToF è simile alla cattura di una fotografia, questi sistemi risultano quindi estremamente veloci. Molti sistemi di visione time of flight raggiungono i 150 frame per secondo.

Tra gli svantaggi di questa tecnologia è importante citare la sensibilità alle altre fonti di illuminazione, definibili come esterne rispetto a quella presente nel sistema ToF e la mutua interferenza nel caso fossero presenti molteplici sensori a tempo di volo inquadranti la medesima scena.

Un esempio di sensore Tof applicato al bin picking è mostrato in Fig. 1.4, si tratta di un sistema *Helios 2* (Lucid vision labs), in questo caso si hanno quattro fonti di illuminazione (diodi VCSEL) e un sensore di immagine (Sony IMX556) con relativa ottica. Essendo dotato di un sensore a colori, questo sistema può fornire una depth map a colori, come mostrato negli esempi di Fig. 1.5.

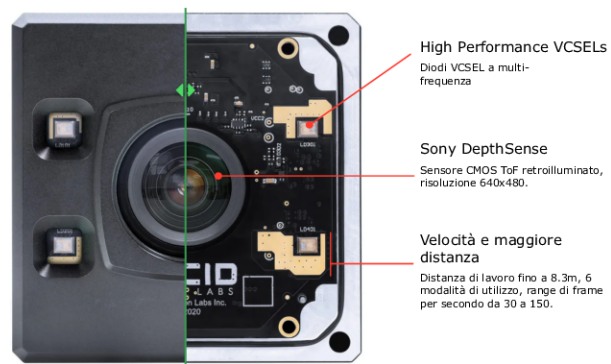


Figura 1.4: Dettaglio del sistema Time of Flight Helios 2 (Lucid vision labs).



Figura 1.5: Acquisizioni dal sistema Time of Flight Helios 2 (Lucid vision labs).

La tecnologia a tempo di volo, oltre a quella appena citata, viene utilizzata in un'ampia varietà di applicazioni:

- **Automotive:** nelle vetture di ultima generazione, questa tecnologia trova applicazione nei sistemi di sicurezza e assistenza alla guida, indicati con l'acronimo *ADAS* (*Advanced Driver Assistance Systems*). In questo campo, i sensori ToF sono principalmente usati per mappare l'ambiente circostante la vettura, in modo che il sistema di controllo al suo interno abbia le informazioni su eventuali ostacoli, altri veicoli o pedoni e a che distanza si trovino questi rispetto alla vettura stessa. Questi dati sono poi utili al sistema di controllo della vettura per funzioni quali il cruise control adattivo, l'assistente per il mantenimento di corsia e l'adattamento automatico dell'altezza dei fari per non abbagliare i veicoli che sopraggiungono in senso opposto. Nella maggior parte dei casi sono impiegati sensori *LiDAR* (*Light Detection and Ranging*), questi sono dei ToF ma con delle leggere differenze. I sensori a tempo di volo come detto utilizzano una sorgente di luce modulata, i sistemi LiDAR invece illuminano la scena con un laser, entrambi si basano sulla misura del tempo trascorso tra generazione della luce e ritorno alla fonte della stessa. Tuttavia, i sensori LiDAR utilizzando un fascio laser, necessitano di un sistema di specchi rotanti, simile a quelli usati nei sistemi a triangolazione laser. La scena di fatto viene scannerizzata, visto che il fascio laser viene fatto muovere, non è un'acquisizione diretta come i ToF descritti in precedenza. Un esempio di sensore ToF per l'automotive, specificamente un LiDAR, è illustrato in Fig. 1.6.

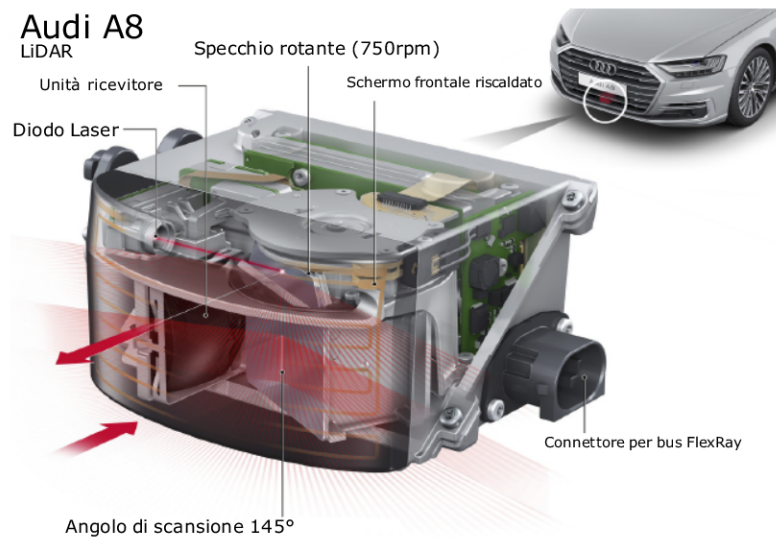


Figura 1.6: Sensore LiDAR installato nei modelli di Audi A8, si noti il sistema di specchi rotanti che permette al fascio laser, generato dal diodo, di scansionare la strada e gli ostacoli di fronte al veicolo.

- **Smartphone:** nei dispositivi di ultima generazione, ad accompagnare il comparto fotografico, vi sono sensori a tempo di volo. La loro funzione è quella di fornire informazioni di profondità al processore di immagine, che in questo modo riesce ad avere migliori performance per la messa a fuoco ed altre funzioni software e post processing di immagini, ad esempio la sfocatura dello sfondo nei ritratti. In questo ambito vengono utilizzati sia sensori Tof a impulsi, come descritto in precedenza, sia sensori LiDAR. In questo modo alcuni smartphone che dispongono di sensori a tempo di volo diventano dei veri e propri scanner 3D, potendo quindi acquisire delle mappe tridimensionali di oggetti o interi ambienti, come illustrato in Fig. 1.7b;

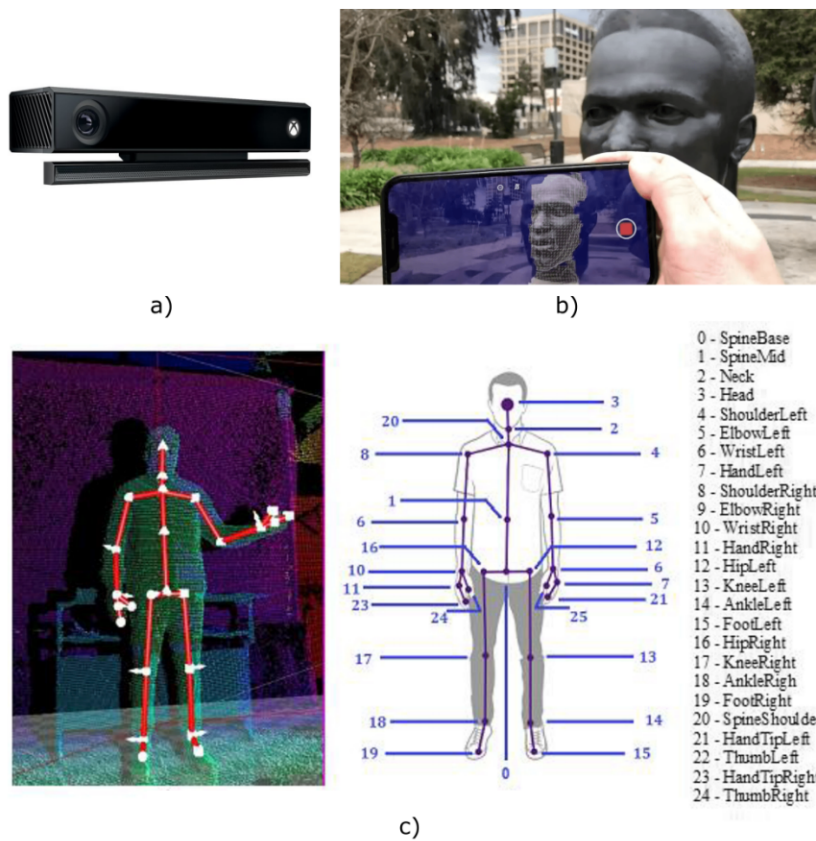


Figura 1.7: a) Microsoft Kinect V2, b) acquisizione 3D di un monumento utilizzando un iPhone 12 Pro, c) Tracking dei movimenti di una persona utilizzando un Kinect V2 come sistema di visione, a sinistra è evidenziato lo *skeleton* rilevato dal sistema, a destra il dettaglio delle sue parti [1].

- **Interfacce uomo macchina e gaming:** grazie all'informazione di profondità, i sistemi a tempo di volo risultano utili anche per tracciare i mo-

vimenti. Per questo trovano applicazione in dispositivi come gli schermi di alcuni laptop ed infotainment delle automobili, rendendo possibile il loro utilizzo con delle *gesture* (movimenti specifici che attivano determinate funzioni) o ancora per rivoluzionare l'interazione ed il controllo nei videogiochi. Il più famoso esempio è probabilmente il *Kinect* di seconda generazione (Fig. 1.7a), un sistema a marchio Microsoft, presente nella confezione della console Xbox One. Questo innovativo accessorio, introdotto nel 2014, include oltre ad un array di microfoni, un sistema a tempo di volo, in grado di rilevare la figura umana e seguire i suoi movimenti come si vede in Fig. 1.7c, permettendo all'utente di interagire con il videogame senza utilizzare il classico controller.

1.2.3 Luce Strutturata

In inglese *structured light*, è uno dei metodi più utilizzati per acquisire immagini di superfici in 3D, i componenti principali sono ancora una volta un dispositivo illuminatore ed un dispositivo sensore. Tuttavia la fonte di luce è diversa da quella utilizzata nelle tecnologie precedentemente descritte, infatti in questi sistemi la scena viene illuminata con un pattern bidimensionale specificatamente progettato che viene fatto variare. Questo è generato da un proiettore o un modulatore di luce, un sensore di immagine è utilizzato per acquisire immagini della scena illuminata dalla appena citata fonte di luce. Il principio di funzionamento dei sistemi di visione a luce strutturata si basa sul rilevare le informazioni riguardanti la distorsione che subisce il pattern di luce proiettato quando incontra una superficie non piana o un oggetto.

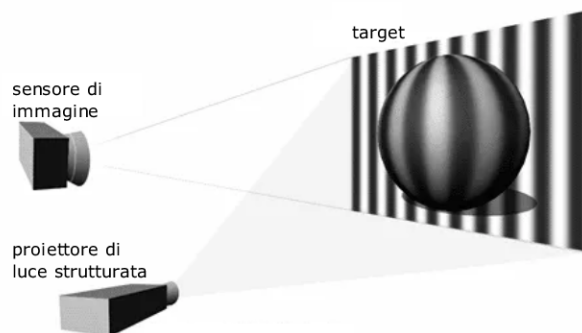


Figura 1.8: Principali componenti di un sistema di visione a luce strutturata. Si noti il pattern proiettato come viene distorto dall'oggetto target presente nella scena.

Basandosi quindi sulla distorsione del pattern di luce, rilevata dal sensore di immagine, rispetto a quello proiettato, è possibile ottenere la forma in 3D degli oggetti presenti nel campo inquadrato. Esistono svariate tipologie di acquisizione

mediante sistemi 3D a luce strutturata, che possono essere divise in due principali categorie:

- *Single-shot*: l'illuminatore proietta un singolo pattern sulla scena, il sensore cattura un'immagine di essa illuminata ed i dati per la mappa 3D sono ottenuti sulla base di quest'ultima. Risulta particolarmente utile nelle applicazioni dove vi sono oggetti in movimento o quando è richiesta una maggiore velocità di acquisizione. Le tecniche a singola acquisizione possono poi essere ulteriormente suddivise in pattern a variazione continua, indicizzazione di striscia e indicizzazione di griglia;
- *Multiple-shot*: anche definite come *sequential*, generalmente consistono nella proiezione di più pattern, in modo sequenziale. Il sensore cattura quindi una molteplicità di immagini, rendendo l'acquisizione più robusta ed accurata. Questa modalità tuttavia essere utilizzata solo in applicazioni con oggetti statici.

Esistono inoltre tecniche ibride, che uniscono alcune caratteristiche delle precedentemente citate. In Fig. 1.9 schema che riassume e illustra graficamente esempi delle tipologie di pattern proiettati.

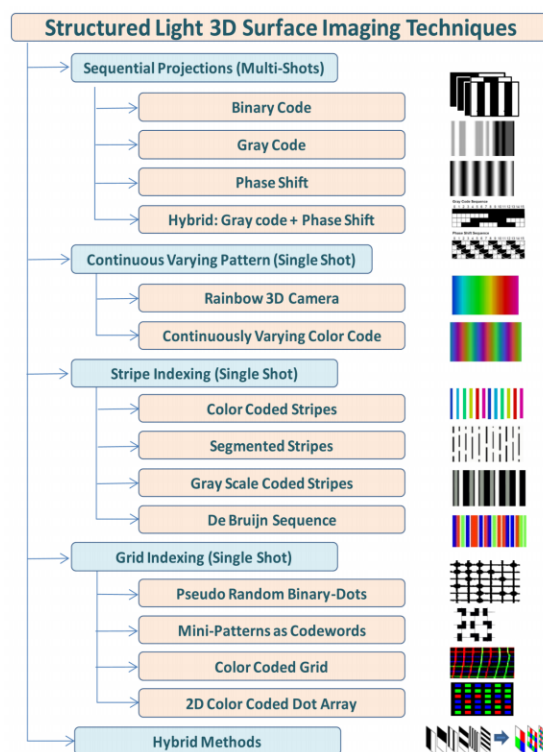


Figura 1.9: Schema rappresentativo delle principali tipologie di pattern proiettati nei sistemi a luce strutturata [2].

Di seguito vengono descritte le tecniche appartenenti alla categoria Multiple shot.

Codifica Binaria (*Binary Coding*)

Questa tecnica utilizza strisce bianche e nere per andare a formare una sequenza di pattern di luce da proiettare, in modo che ogni punto della superficie dell'oggetto abbia un codice binario univoco, differente dal codice di un qualsiasi altro punto nella scena. Avendo a disposizione, a titolo di esempio, un pattern a 5 bit, quando proiettato si ottengono $2^5=32$ aree codificate univocamente. Mediante il principio di triangolazione è possibile calcolare le coordinate 3D (x,y,z) per tutti e 32 i punti lungo ogni linea orizzontale, ottenendo così l'intera immagine 3D. In Fig 1.10 sono raffigurati in basso i pattern proiettati sequenzialmente, ed in alto il risultato di tale proiezione sulla scena. Come si nota, la scena viene suddivisa in 32 linee, ognuna di esse possiede una codifica univoca, diversa dalle altre. Tra i vantaggi principali di questa tecnica si ha l'affidabilità e la poca sensibilità alle caratteristiche della superficie analizzata. Tuttavia, dato che è richiesta la proiezione di molteplici pattern, la durata temporale dell'acquisizione risulta elevata, tanto da rendere la codifica binaria non adatta per alcune applicazioni. Infine è richiesto che la scena non cambi durante questo lasso di tempo.

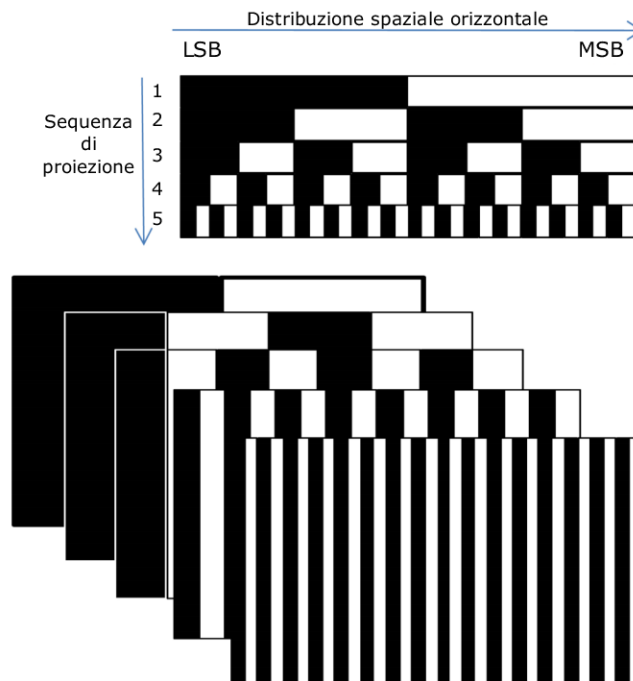


Figura 1.10: Sequenza di pattern di luce proiettati da un sistema di visione a luce strutturata basato su codifica binaria a 5 bit [2].

Pattern a scala di grigi (*Gray-Level Patterns*)

Questa tecnica nasce con l'obiettivo di ridurre il numero di pattern da proiettare, è infatti molto simile alla precedentemente descritta codifica binaria, la differenza sta proprio nella tecnica di illuminazione: ogni singolo pattern viene proiettato con distinti livelli di intensità, passando da una codifica di 2^N strisce, con N numero di pattern, a una con M^N strisce, con M numero di livelli di intensità del singolo pattern. Di fatto, i pattern sono i medesimi, ma vengono proiettati con diversi livelli di intensità luminosa. Se con la tecnica *binary coding* per ottenere 64 aree sono necessari 6 pattern ($2^6=64$), con la *gray-level* sono sufficienti 3 pattern ognuno eseguito con 4 differenti livelli di luminosità ($4^3=64$). In generale, il criterio per la scelta dei valori di M ed N si concentra sulla massimizzazione della risoluzione ad una definita distanza di misura.

Sfasamento (*Phase Shift*)

In questa tecnica vengono proiettati sulla scena un set di pattern sinusoidali, ogni pixel di questi pattern è descritto come

$$\begin{aligned} I_1(x, y) &= I_0(x, y) + I_{mod}(x, y)\cos(\phi(x, y) - \theta) \\ I_2(x, y) &= I_0(x, y) + I_{mod}(x, y)\cos(\phi(x, y)) \\ I_3(x, y) &= I_0(x, y) + I_{mod}(x, y)\cos(\phi(x, y) + \theta) \end{aligned} \quad (1.3)$$

Dove I rappresenta l'intensità dei pattern, I_0 è la componente costante, anche definita DC, I_{mod} è la modulazione dell'ampiezza del segnale, ϕ è la fase e θ l'angolo di sfasamento, costante. In Fig. 1.11 è illustrato un esempio dei tre segnali appena descritti, ed un pattern luminoso ottenuto mediante questa tecnica, a cui ci si riferisce con il nome di *fringe image*. Per ottenere il valore dello sfasamento $\phi(x,y)$ è necessario calcolare dapprima $\phi'(x,y)$ come segue

$$\phi'(x, y) = \arctan \left[\sqrt{3} \frac{I_1(x, y) - I_3(x, y)}{2I_2(x, y) - I_1(x, y) - I_3(x, y)} \right] \quad (1.4)$$

Questo perché la funzione arcotangente presenta discontinuità in corrispondenza di 2π , per ottenere il valore di ϕ è sufficiente sommare multipli di 2π al precedentemente calcolato $\phi'(x,y)$.

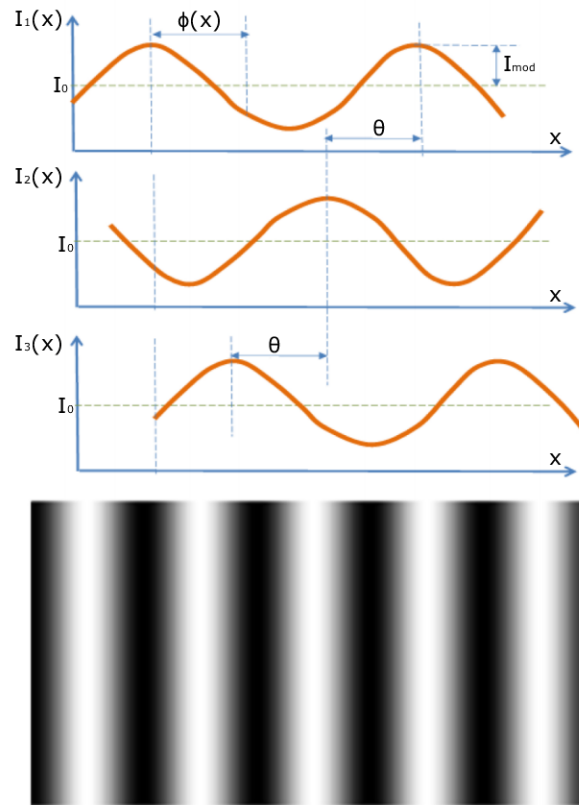


Figura 1.11: In alto, un esempio di tre segnali sinusoidali modulati. In basso, il pattern di luce risultante utilizzando tali segnali in un sistema di visione a luce strutturata basato sulla tecnica del phase shift [2].

Per ottenere le coordinate (x,y,z) è necessario basarsi sulla differenza tra la fase misurata ϕ ed il valore di fase rispetto ad un piano di riferimento. Utilizzando la configurazione di Fig 1.12 si ha la seguente relazione

$$\frac{Z}{L-Z} = \frac{d}{B} \quad (1.5)$$

quindi la coordinata relativa all'altezza è

$$Z = d \frac{L-Z}{B} \quad (1.6)$$

che semplificata diventa

$$Z \approx \frac{L}{B} d \propto \frac{L}{B} (\phi - \phi_0) \quad (1.7)$$

dove ϕ_0 è la fase rispetto al piano di riferimento.

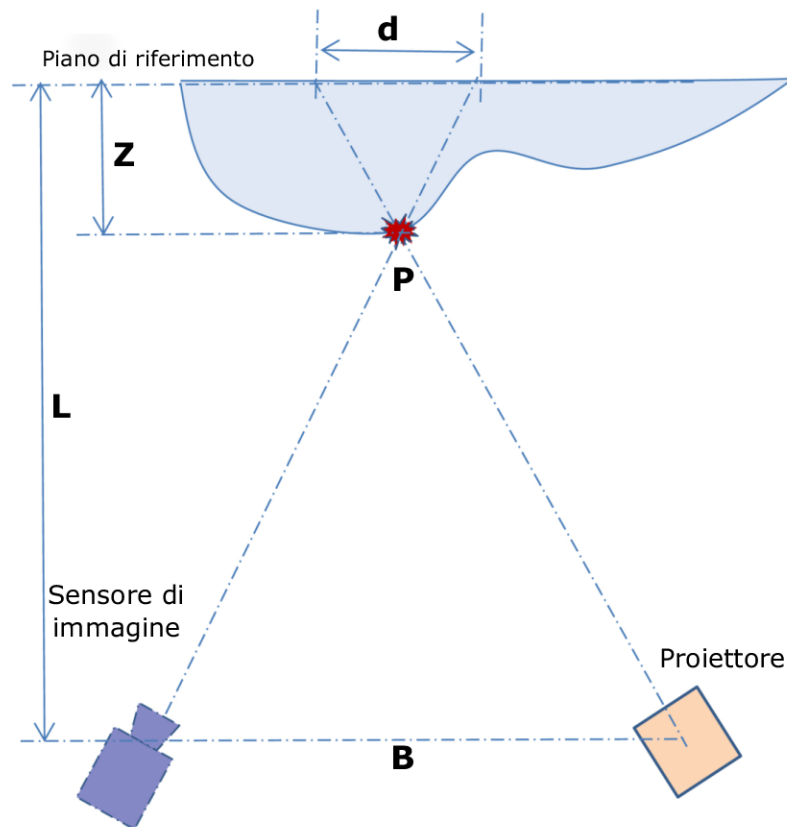


Figura 1.12: Esempio di configurazione per un sistema di visione a luce strutturata.

I principali vantaggi dei sistemi di visione a luce strutturata sono:

- **Risoluzione:** è possibile ottenere risoluzioni estremamente elevate, dell'ordine dei micron, per volumi ridotti, e in alcuni casi risoluzioni ridotte ma per volumi elevati. Risulta quindi una tecnologia flessibile;
- **Sistema fisso:** il sistema telecamera-proiettore non è necessario venga spostato per ottenere la scansione 3D;
- **Velocità:** rispetto ad altre tecnologie, non è necessario eseguire una scansione vera e propria della scena, come accade con le tecnologie laser descritte precedentemente. È sufficiente proiettare i pattern e catturare le immagini, questo risulta in un tempo di acquisizione ridotto.

Tra gli svantaggi di questa tecnologia è importante citare:

- **Sensibilità a riflessioni e luce ambientale:** oggetti riflettenti o trasparenti risultano particolarmente problematici, in quanto i pattern proiettati vengono riflessi quindi le telecamere rilevano una distorsione di questi che non

rappresenta la reale conformazione della scena. Questo causa assenza di dati nelle acquisizioni o in generale mappe 3D non rappresentative. Le fonti di illuminazione presenti nella scena, come ad esempio quella solare che filtra attraverso una finestra, rappresentano fonte di disturbo per il sistema a luce strutturata perché vanno a interferire con il pattern proiettato;

- **Necessità di più telecamere:** nelle maggior parte delle applicazioni sono richieste due o più telecamere, per far fronte al problema delle occlusioni. Questo si presenta quando un oggetto nella scena copre il pattern proiettato, rendendo impossibile ottenere dati sul volume che questo oggetto copre. La parte di scena che risulta in ombra non ha quindi dati, creando dei veri e propri *buchi* nella mappa 3D. Per risolvere questo problema, i sistemi di visione a luce strutturata includono due o più telecamere, per ottenere immagini dalla scena a diverse angolazioni. Questo si traduce in maggiore complessità, costo e dimensioni;
- **Complessità computazionale:** per ottenere risultati con risoluzioni elevate è richiesto un elevato numero di immagini catturate dai sensori, che poi vengono processate. Sono quindi richiesti processori con una buona potenza computazionale.

In aggiunta, come citato in precedenza, la scena deve essere statica e priva di alcun tipo di vibrazione, che renderebbero altrimenti le acquisizioni distorte. La tecnologia a luce strutturata è una delle più utilizzate per la visione 3D, un esempio applicato al bin picking industriale è illustrato in Fig. 1.13 dove è riportato un sistema *Fanuc*, più precisamente il *3D Area Sensor*.

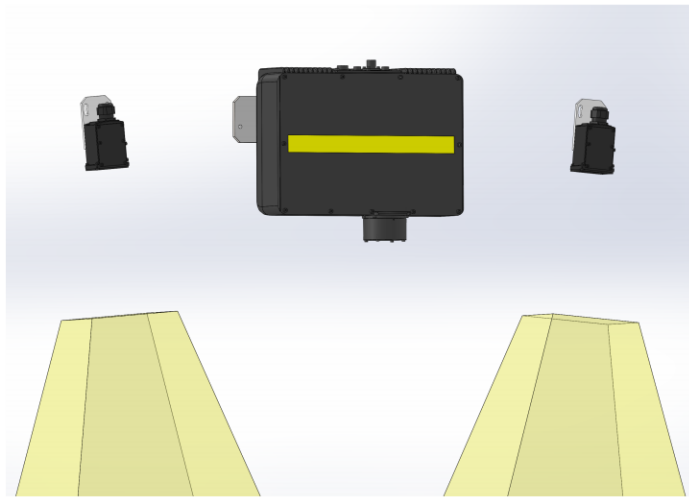


Figura 1.13: Sistema di visione Fanuc 3D Area Sensor, composto da un proiettore (al centro) e due telecamere (a destra e sinistra del proiettore). In giallo una parte del campo inquadrato da quest'ultime.

Un altro esempio è riportato in Fig. 1.14 dove viene rappresentato il *Keyence serie CV-X*, modello RB-1200. Questo è un sistema di visione avanzato ed estremamente performante, come si può vedere nella stessa figura, è composto da un proiettore e quattro sensori di immagini, al fine di rendere lo stesso più robusto rispetto alle appena discusse criticità tipiche di questa tecnologia.

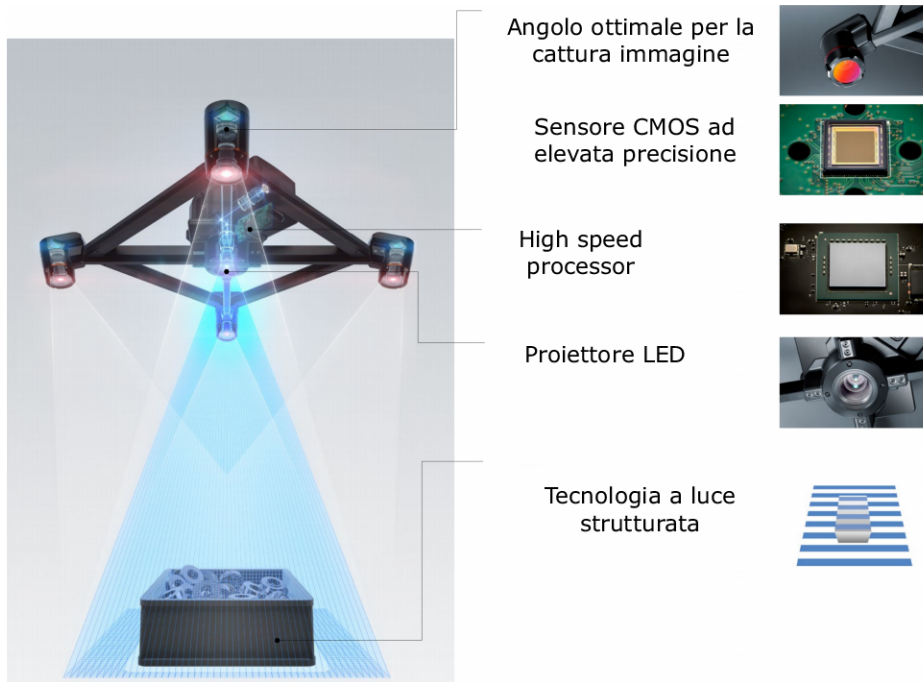


Figura 1.14: A sinistra, un sistema di visione *Keyence serie CV-X*, modello RB-1200. A destra, le sue principali caratteristiche.

1.2.4 Visione Stereoscopica

A differenza delle tecniche descritte in precedenza, questa appartiene alla categoria dei metodi ottici passivi, sfrutta quindi un'illuminazione passiva, quella dell'ambiente. Generalmente, si ottengono acquisizioni 3D meno robuste e dense rispetto a quelle prodotte da sistemi a illuminazione attiva. Nello specifico questa tecnica di acquisizione 3D fa uso di due telecamere che inquadrano la medesima scena di interesse da due differenti punti di vista, si basa sempre sul principio della triangolazione. Secondo quest'ultimo, nota la geometria delle due telecamere è possibile triangolare le coordinate tridimensionali di un punto della scena. La visione stereoscopica, a cui ci si riferisce anche come *stereo imaging*, si compone dei seguenti step:

- *Underdistortion*: rimozione matematica della distorsione radiale e tangenziale delle lenti, ottenendo quindi un'immagine non più distorta;

- *Rectification*: regolazione secondo gli angoli e le distanze tra le telecamere;
- *Correspondance*: rilevazione delle stesse caratteristiche nella vista di una e dell'altra telecamera. Si ottiene una *disparity map* ossia una mappa delle differenze, dove queste sono in coordinate x sul piano delle immagini, riferite alle stesse caratteristiche viste dalle telecamere;
- *Reprojection*: nota la disposizione delle telecamere, è possibile tradurre la disparity map in distanze, mediante la triangolazione. Si ottiene quindi una depth map, mappa che contiene le informazioni di profondità.

La geometria generica di una coppia di telecamere stereo (qui semplificate secondo il modello *pinhole*) è presentata in Fig 1.15 e prende il nome di *geometria epipolare*. O_l e O_r sono i due centri di proiezione, i segmenti che li congiungono a P sono detti raggi proiettivi e intersecano i due piani immagine rispettivamente nei punti p_l e p_r . I due punti e_l e e_r sono detti *epipoli*, e sono rispettivamente la proiezione di O_r sul piano immagine della telecamera di sinistra e la proiezione di O_l sul piano immagine della telecamera di destra. Le rette tratteggiate che congiungono p_l a e_l e p_r a e_r sono dette *rette epipolari*. La retta epipolare della telecamera di sinistra è l'immagine del raggio proiettivo che collega P a O_r e viceversa. Di conseguenza, data la proiezione di un punto su una delle due immagini, la corrispondenza sull'altra immagine è situata lungo la relativa retta epipolare. Questa proprietà consente di semplificare notevolmente il calcolo delle corrispondenze.

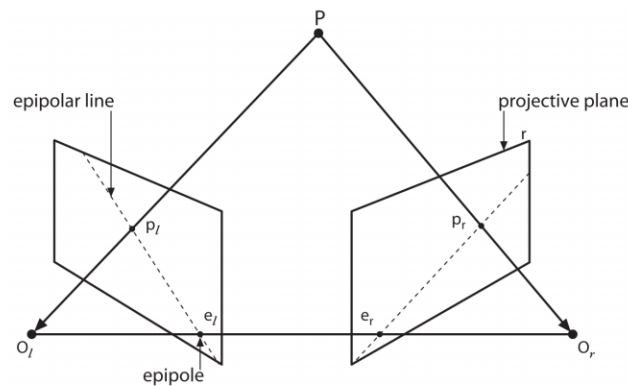


Figura 1.15: Generica geometria per una coppia di telecamere, si noti il piano epipolare definito dal punto osservato P e dai due centri di proiezione O [3].

A valle della precedentemente citata operazione di *Rectification*, si ottiene una geometria ideale che ha la particolarità di avere entrambi i piani immagine coplanari e con le righe allineate, come illustrato in Fig 1.16. Il sistema di riferimento della scena è centrato su O_l centro di proiezione della camera di sinistra, e i due assi principali sono paralleli all'asse Z, intersecando i piani immagine nei rispettivi punti principali. Si assume che la distanza f del piano immagine dal centro di

proiezione sia la medesima per entrambe le telecamere. La differenza tra le due coordinate x dei punti p_l e p_r è definita *disparità* come citato in precedenza. Si noti che in questo caso le rette epipolari corrispondono alle righe dell'immagine, pertanto punti coniugati possiedono il medesimo indice di riga. Dopo aver individuato le corrispondenze nelle due immagini rettificate, è sufficiente proiettare nuovamente le coordinate dal piano immagine al sistema di riferimento della scena, si nota che la quota di P lungo la direzione Z è inversamente proporzionale al valore di disparità. Tale quota $Z = \frac{f \cdot T}{d}$ nella quale d è la disparità, definita come $d = x_l - x_r$. Nel caso più generale, in cui le coordinate dei punti principali differiscono la quota Z è $Z = \frac{f \cdot T}{d - (c_x^l - c_x^r)}$.

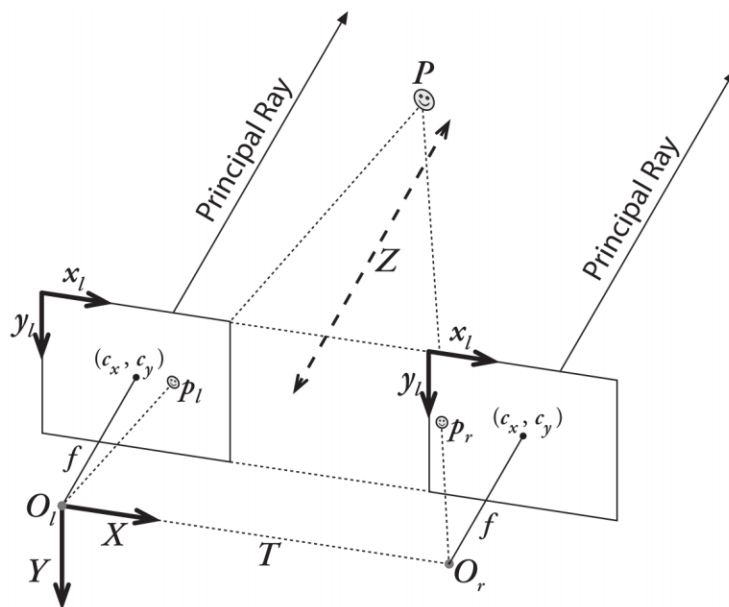


Figura 1.16: Sistema di coordinate per telecamere non distorte e rettificate. Le coordinate dei pixel sono relative all'angolo superiore sinistro dell'immagine, i due piani sono allineati [3](#).

Capitolo 2

Cella robotizzata in laboratorio

2.1 Introduzione

Nel seguente capitolo viene dapprima descritto il setup presente in laboratorio, composto da robot e sistema di visione ad esso collegato. Successivamente si descrive la logica per il bin picking predisposta da Fanuc, con le relative funzionalità software. In questo documento, tutti i test e le rilevazioni eseguite nei Cap.3 e Cap.4 sono eseguiti su un sistema composto da un robot antropomorfo ed un sistema di visione 3D entrambi a marchio *Fanuc*.

2.2 Robot

Il manipolatore utilizzato è un *Fanuc CR-15iA*, ha la caratteristica di essere un robot antropomorfo di tipo collaborativo e dispone di 6 gradi di libertà. In Fig.2.1 è riportato un estratto dal datasheet per illustrare lo spazio di lavoro del robot. Il carico che può sostenere al polso (*payload*), è di 15Kg e la massima velocità dell'organo terminale è di 800mm/s. Il robot è definito collaborativo, quindi in grado di interagire con l'operatore, per questo non necessita di barriere di sicurezza. Questo perché sul manipolatore è presente una serie di sensori in grado di rilevare l'eventuale collisione con l'operatore, riducendo la possibilità di recare danni allo stesso. In particolare, il modello in esame dispone di sensoristica installata all'interno della base, in aggiunta è stato montato un sensore di forza sul polso del robot, questo si trova tra il polso e l'organo terminale ed è un sensore *Fanuc FS-15iA*. Risulta utile nelle applicazioni che richiedono particolare attenzione alla forza applicata dall'organo terminale, ad esempio per assemblaggi e lucidature, o più banalmente dove è necessario pesare gli oggetti movimentati. L'organo terminale installato è una pinza *Schunk Co-act EGP-C*, si tratta di una pinza collaborativa, elettroattuata, con griffe parallele. Questa ha un'apertura minima di 25mm ed un'apertura massima di 40mm, la forza di presa massima è di 140N ed è regolabile su quattro differenti livelli (25%, 50%, 75% e 100% del valore massimo). Per interfacciarsi con il controllore (*Fanuc R-30iB*) del robot

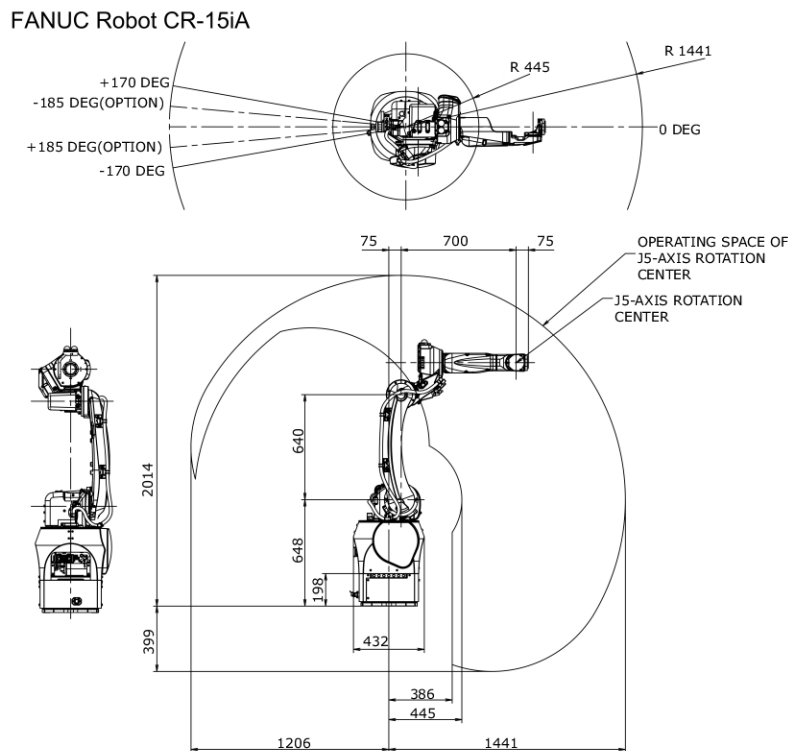


Figura 2.1: Workspace del robot Fanuc CR15-iA.

è predisposta una connessione via Ethernet. Nel controllore utilizzato è stato impostato l'indirizzo IP 192.168.1.1, è sufficiente collegare un PC al controllore stesso tramite cavo Ethernet, impostando un IP statico sulla scheda di rete del computer, che appartenga alla medesima rete e digitare poi l'appena citato indirizzo IP su Internet Explorer (oppure Microsoft Edge in modalità Explorer). La finestra visualizzata è la *ROBOT Homepage* (Fig. 2.2), tramite la quale è possibile accedere a tutte le principali funzioni disponibili nel controllore.

2.3 Sistema di visione

Nella cella robotizzata in esame, ad affiancare il robot appena descritto, vi è un sistema di visione tridimensionale *Fanuc 3D Area Sensor*, a cui vi si riferisce anche come *3DA-1300*. Come illustrato in Fig. 2.3a è composto da un proiettore e due telecamere, poste alla destra e alla sinistra del primo, è stato già citato nel Cap. 1 in Fig. 1.13 in quanto facente parte dei sistemi di visione a luce strutturata. Come trattato in tale capitolo, questo sistema ottiene le informazioni 3D all'interno del suo campo visivo proiettando diversi pattern a strisce e acquisendo immagini con le due telecamere. Si tratta di un sistema fisso, quindi che va installato su una struttura solida, come illustrato in Fig. 2.3b. Le telecamere

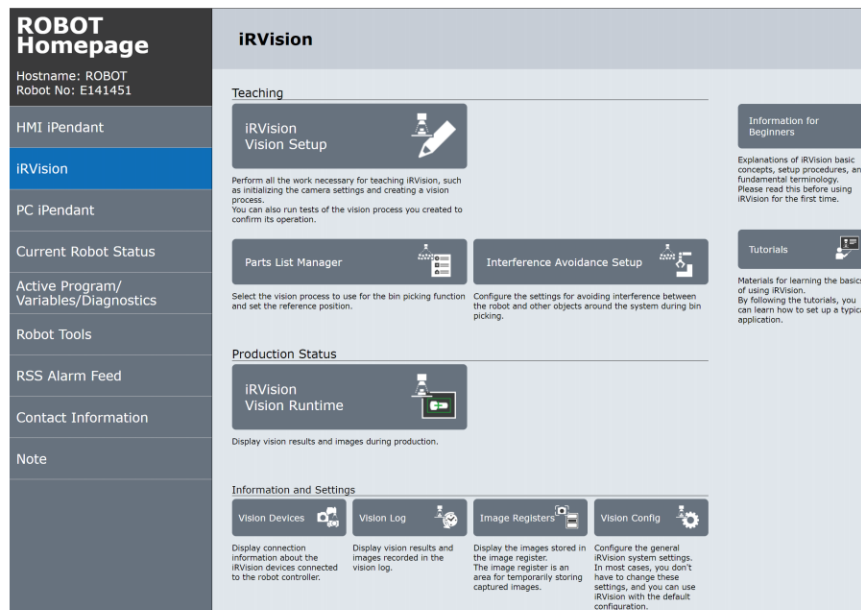


Figura 2.2: Finestra ROBOT Homepage del controllore Fanuc.

sono in bianco e nero ed hanno una risoluzione di 1024x1280 pixel. Il sistema viene indicato anche come 3DA-1300 perché ha la caratteristica di avere un campo inquadrato di dimensioni considerevoli, 1300x1000x1000mm. Questo, per le dimensioni della base, corrisponde ad un euro pallet, risulta quindi particolarmente adatto ad applicazioni di depallettizzazione. Di particolare importanza è la definizione di tale campo inquadrato (*FOV Field Of View*): il sensore 3DA calcola un certo numero di punti all'interno del campo inquadrato dal proiettore (se in modalità *Normal density* sono 239x192), per questo la risoluzione o densità spaziale dei punti 3D misurati dipende dal FOV del proiettore stesso. Più questo è ampio, più i punti saranno distanziati tra loro, dato che il pattern proiettato è meno intenso, di conseguenza, non potendo aumentare la luminosità del proiettore essendo fissa, è consigliato che il FOV dello stesso sia il più piccolo possibile. Per quanto riguarda il campo inquadrato dalle due telecamere, come il precedente FOV deve includere l'intero contenitore da cui verranno prelevati i pezzi ed in generale deve includere l'area inquadrata dal proiettore. Infine, perché le informazioni di posizione ed orientazione rilevate dal sistema di visione possano essere utilizzate dal robot, è necessario convertirle dal sistema di riferimento *visione* al sistema di riferimento *robot*. Per fare ciò, il robot deve conoscere la posizione del sistema di visione rispetto al suo sistema di riferimento, è necessario quindi eseguire una *calibrazione*. Questa può essere:

- grid pattern calibration: nel campo inquadrato dal sistema di visione si posiziona una griglia di calibrazione, fornita da Fanuc. Il sistema si calibra autonomamente acquisendo svariate immagini di tale griglia;

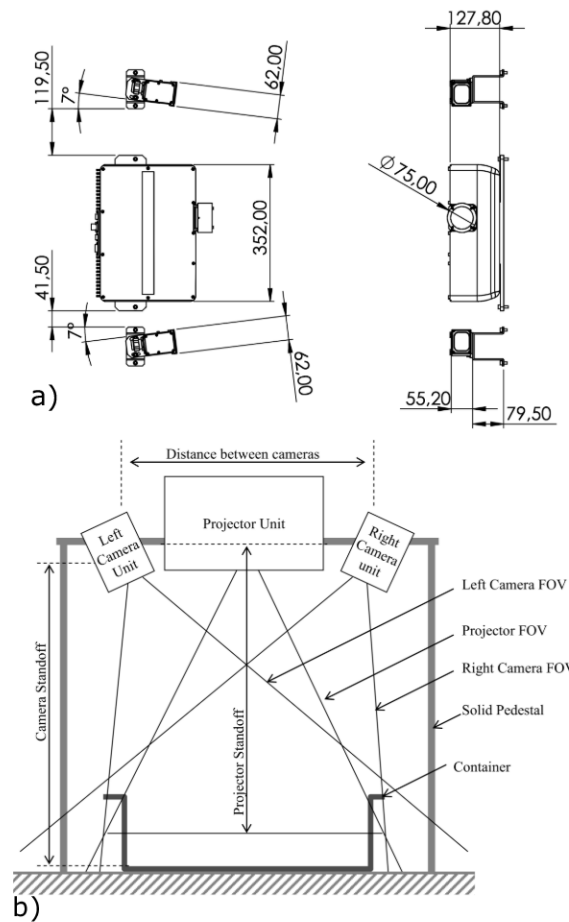


Figura 2.3: a) sistema di visione Fanuc 3D Area Sensor, b) disposizione dello stesso [4](#).

- robot generated grid calibration: prevede di installare un target sulla flangia del robot, il quale muove tale target secondo uno specifico percorso nel campo inquadrato dal sistema di visione. Questo acquisisce diverse immagini e riconoscendo il target ottiene le informazioni necessarie alla calibrazione.

Il metodo di calibrazione utilizzato è il secondo, in quanto consigliato da Fanuc, visto che permette di ottenere un'accuratezza maggiore. La cella robotizzata in esame (Fig [2.4](#)) si compone dei precedentemente descritti robot e sistema di visione, comprensivo di struttura, del controllore del robot stesso e di un tavolo avente la funzione di piano di lavoro. Su di esso è poggiato il contenitore (o i contenitori) dei pezzi da movimentare.

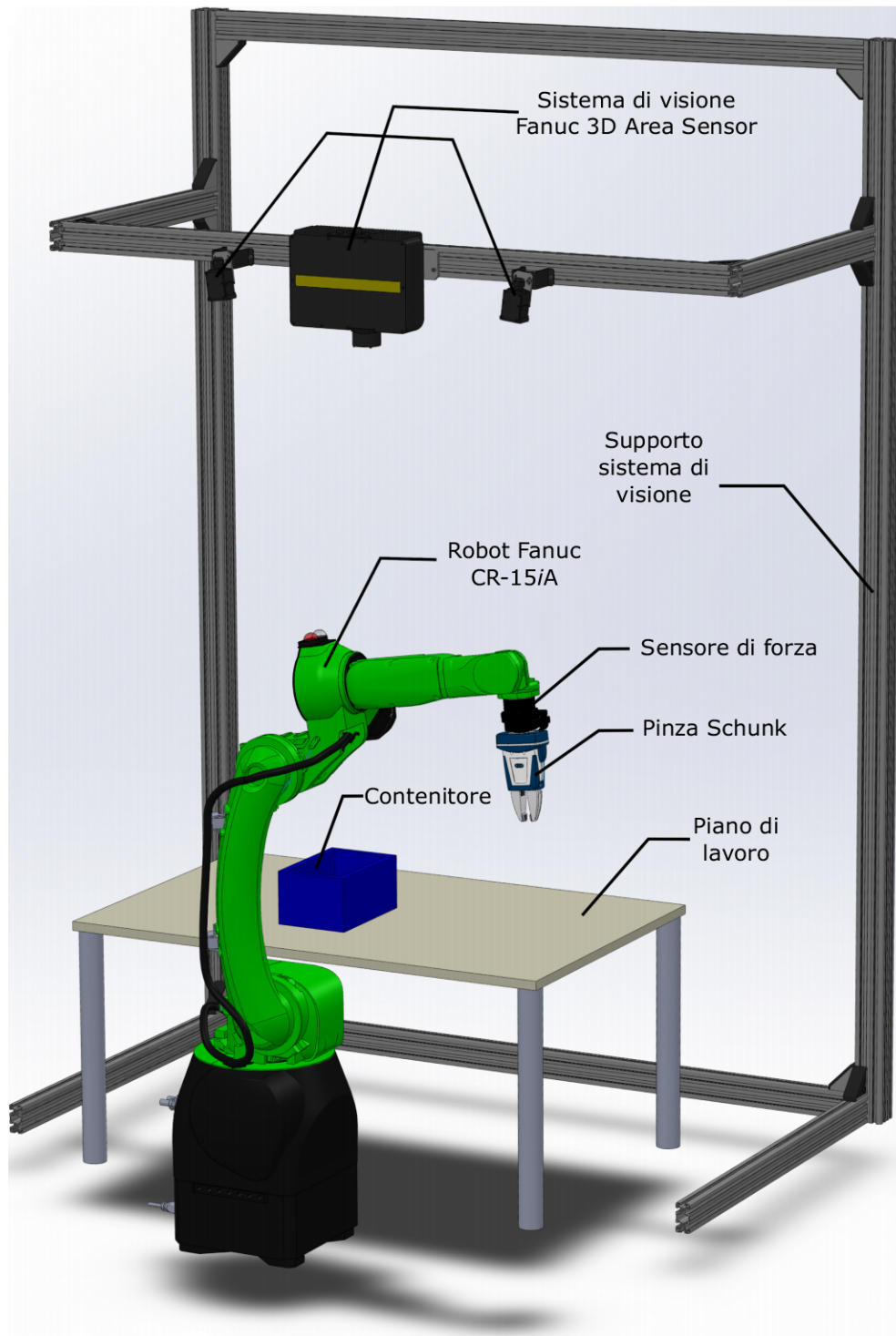


Figura 2.4: Cella robotizzata presente in laboratorio.

2.4 Fanuc Bin Picking

Come descritto nel Cap 1, il bin picking è un'applicazione nella quale viene riconosciuta posizione ed orientazione di componenti disposti casualmente all'interno di un contenitore, il robot poi preleva tali componenti uno ad uno, seguendo quindi il flusso indicato in Fig 2.5a. Nel sistema Fanuc descritto nella sezione precedente, il bin picking viene implementato utilizzando due funzioni di base messe a disposizione dal costruttore:

- *Parts List Manager*;
- *Interference Avoidance*.

Queste due fondamentali funzionalità fanno parte del pacchetto *Fanuc iRVision*. In aggiunta è necessario impostare un sistema di riferimento *User Frame*, che sarà utilizzato in seguito per i calcoli di *Interference Avoidance*, ed uno per la pinza *Tool Frame* come indicato in Fig 2.5b.

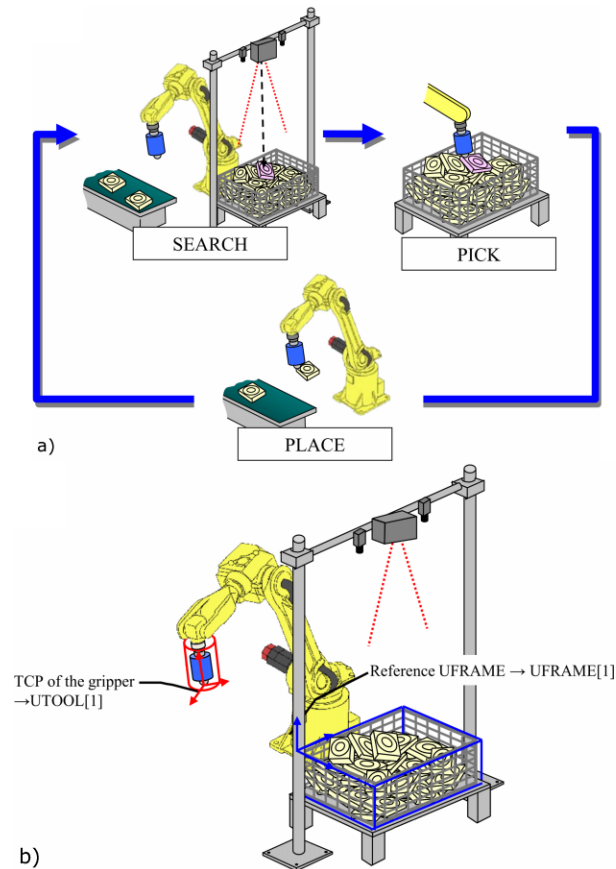


Figura 2.5: a) flusso del processo di Bin Picking, b) tool frame e user frame da impostare 4.

2.4.1 Parts List Manager

Il *Parts List Manager* è un insieme di funzioni necessarie per il Bin Picking di Fanuc, la logica di funzionamento è composta da quattro step ed è riportata in Fig. 2.6

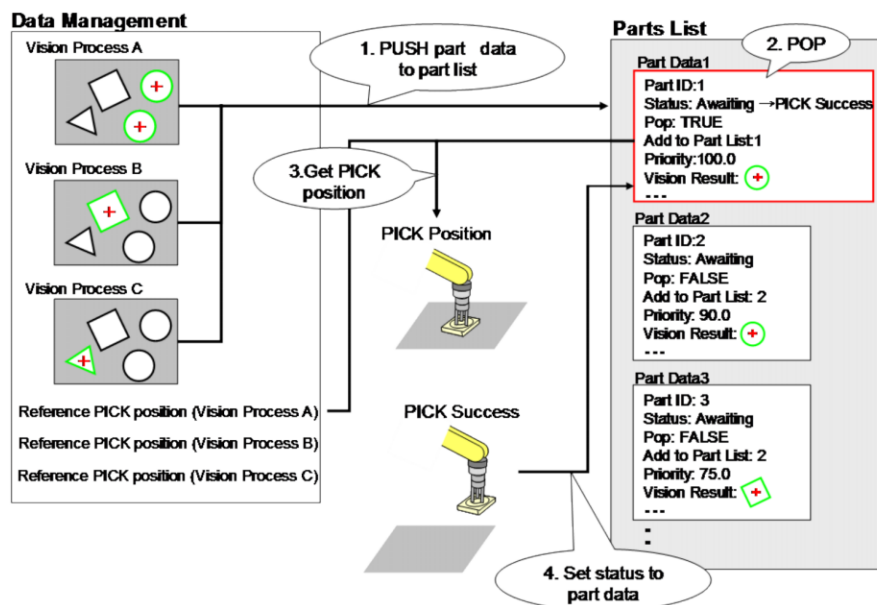


Figura 2.6: Logica di funzionamento di un processo di Bin Picking utilizzando il Fanuc Parts List Manager [4].

1. **Push:** un processo *SEARCH* indica il rilevamento di componenti all'interno di un contenitore, a seguito di un'acquisizione eseguita con il 3D Area Sensor. Come output si ottengono posizione ed orientazione di ogni componente riconosciuto. Questi dati, per ogni componente, vanno a comporre un insieme di dati a cui ci si riferisce con il nome di *Part Data*, e ad ognuno di questi viene associato un numero identificativo univoco, il part data ID. Lo step di Push crea tali Part Data e li aggiunge alla Parts List, che rappresenta una collezione di tali tipi di dati;
2. **Pop:** dalla lista appena citata, viene selezionato un candidato, che sarà il componente da prelevare dal contenitore, sulla base di una priorità che è definita dall'utente mediante un insieme di valori, fino a dieci (ad esempio il componente che sopra tutti gli altri, ossia quello avente coordinata Z più alta e che ha il punteggio di riconoscimento più alto);
3. **Ottenimento della posizione di Pick:** a seguito del Pop, si hanno a disposizione i dati dello specifico e scelto componente nel contenitore, più precisamente il *vision offset data*. Utilizzando questo dato e la posizione di

riferimento del robot, viene calcolata la posizione alla quale il robot deve portarsi per eseguire la presa del pezzo. Questa viene salvata in un registro di posizione (*PR Position Register*);

4. **Impostazione dello status:** quando viene eseguita una qualsiasi operazione sul componente di cui è stato fatto il pop, viene modificato lo status che indica lo stato del componente stesso. In questo modo viene fatto coincidere lo stato del reale componente con lo stato del medesimo presente nella Parts List.

I quattro step appena descritti vengono gestiti mediante un ulteriore set di funzioni:

- **Data Management function:** si occupa di gestire i dati ed imposta un processo di visione come attivo oltre alle posizioni di riferimento utilizzate;
- **Position Get function:** permette al Parts List Manager di calcolare automaticamente le posizioni, la più importante è la posizione di PICK ossia quella per prelevare il componente dal contenitore, e di salvarla in un registro di posizione.

2.4.2 Interference Avoidance

La funzione di *Interference Avoidance* è, come detto, una funzione di base per il Bin Picking, permette di gestire ed evitare le collisioni durante un processo e nel sistema Fanuc è composta dalle seguenti tre funzioni:

- **Interference Check:** controlla se si verificano collisioni tra il tool montato sul robot e gli oggetti presenti nelle vicinanze dello stesso;
- **Interference Avoidance:** in aggiunta alla precedente funzione, questa calcola automaticamente una posizione ed orientazione che possa permettere al robot di muoversi, evitando di collidere con l'ambiente circostante;
- **Wall Avoidance:** calcola una quota perché il robot ritragga il tool dalle pareti del contenitore verso il centro dello stesso. Questa viene utilizzata dopo un PICK.

Per utilizzare le funzioni appena citate è necessario impostare la posizione e la dimensione degli oggetti sui quali il robot farà il controllo delle collisioni. Questi nel sistema Fanuc sono divisi in due distinte categorie:

- *Interference Avoidance Setup System:* ne fanno parte gli oggetti presenti nell'ambiente circostante al robot, quali ad esempio il supporto del sistema di visione ed il contenitore dei componenti da movimentare. Tutti gli oggetti di questo tipo, a parte il contenitore, sono definiti oggetti fissi (*fixed objects*). Il setup della posizione e delle dimensioni del container

prevede l'utilizzo del *three point method*, è necessario quindi portare il tool del robot in tre specifici punti e registrare le relative coordinate. I punti sono quelli indicati in Fig 2.7a, a cui si aggiunge anche la profondità del contenitore. Gli altri oggetti nell'ambiente circostante al robot possono essere rappresentati mediante cilindri, sfere ed esadri come indicato in Fig 2.7b,c,d;

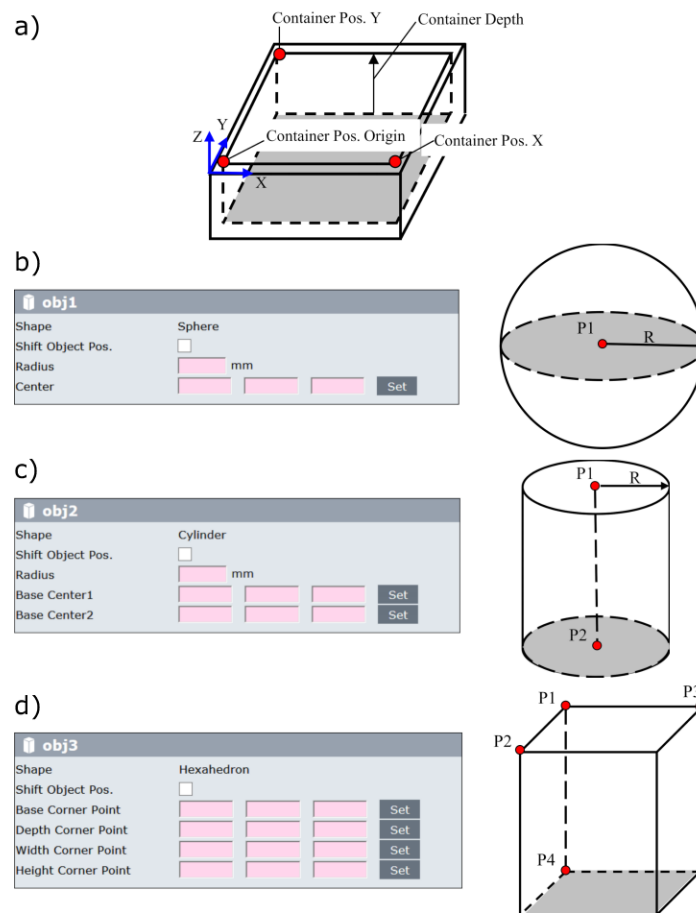


Figura 2.7: a) punti necessari per impostare dimensioni e posizione del contenitore, b,c,d) poliedri per rappresentare gli oggetti nelle vicinanze del robot, nell'Interference Avoidance [4].

- *Interference Avoidance Setup Robot*: ne fanno parte gli oggetti montati sul robot, nella maggior parte delle applicazioni questi sono la pinza ed i relativi cablaggi. Anche in questo caso è possibile utilizzare delle forme standard (cilindro, sfera ed esadro) al fine di rappresentare il tool per il calcolo delle collisioni. Nell'impostare la forma del tool è fondamentale selezionare *HAND* come 'Type', questa sarà infatti la forma che il robot

utilizzerà per fare il controllo delle collisioni all'interno del contenitore quindi per far funzionare correttamente l'Interference Avoidance.

Un esempio di cella nella quale sono stati impostati i sopra descritti setup è riportata in Fig 2.8) dove il contenitore è poggiato su un tavolo, inserito utilizzando la forma esadro, ed il tool è rappresentato anch'esso da un esadro.

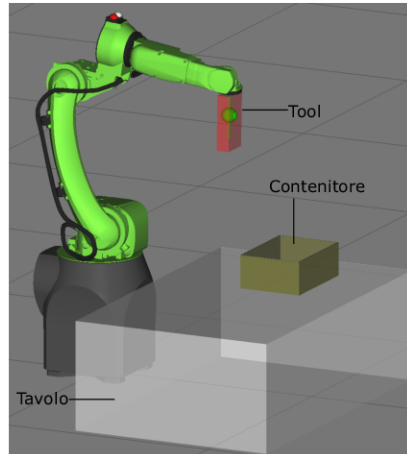


Figura 2.8: Esempio di cella in cui sono stati inseriti i dati degli oggetti per l'Interference Avoidance.

Le condizioni rispetto alle quali fare il controllo delle collisioni vengono invece raccolte nella categoria *Interference Avoidance Setup Condition*, in cui sono disponibili le impostazioni di Fig 2.9). In questa funzione, mediante la voce 'Type', è possibile selezionare che tipo di Interference Avoidance utilizzare, tra le tre descritte ad inizio della Sez 2.4.2). Il 'Utool Number' deve coincidere con il Tool

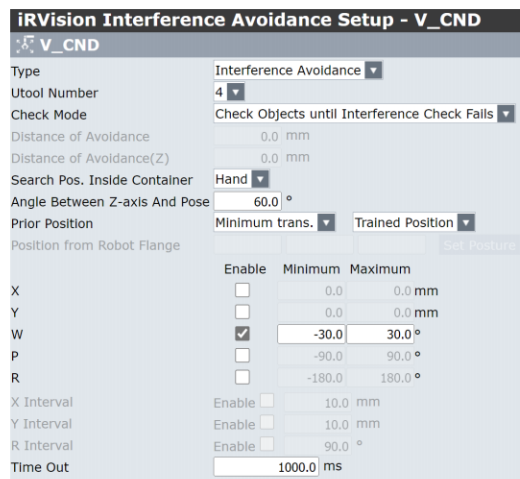


Figura 2.9: Interfaccia di settaggio dell'Interference Avoidance Setup Condition.

frame impostato per il gripper che si utilizzerà nell'applicazione finale, alla voce 'Check Mode' è possibile scegliere il comportamento del robot nel caso di rilevazione di interferenza. Di fondamentale importanza è la voce 'Search Pos. Inside Container', dove è necessario selezionare HAND in modo che le collisioni siano rilevate sulla base del modello definito come HAND nell'Interference Avoidance Setup Robot. In questo modo la configurazione calcolata per evitare la collisione sarà sempre una configurazione interna al contenitore. I parametri successivi si riferiscono a come viene calcolata tale configurazione alternativa, nello specifico:

- *Angle Between Z-axis And Pose*: si riferisce al limite sull'angolo tra l'asse Z dello User frame e l'asse Z del Tool frame, come indicato in Fig 2.10. La posizione ed orientazione alternativa calcolata dal sistema avrà un angolo α che non supera il valore impostato. Nel caso in cui non fosse possibile ottenere un'orientazione che rispetti tale limite, il calcolo è giudicato fallito;

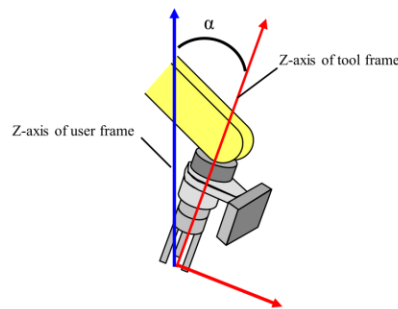


Figura 2.10: Rappresentazione dell'angolo α tra asse Z dello User frame ed asse Z del Tool frame [4].

- *Prior Position*: in questo parametro è possibile selezionare il metodo per il calcolo della posizione ed orientazione di Interference Avoidance, ed inoltre il metodo di selezione tra quelle calcolate. Come metodo di calcolo si hanno due possibilità: *Minimum Trans* che calcola la nuova posizione ed orientazione andando a minimizzare lo spostamento in X ed Y dell'end effector, e *Minimum R* che minimizza la rotazione del tool rispetto all'asse Z dello stesso. In base al metodo di calcolo selezionato si hanno a disposizione diversi metodi di selezione, come indicato in Fig 2.11. Quindi tra le posizioni ed orientazioni calcolate nell'Interference Avoidance, viene scelta quella che meglio soddisfa il metodo di sorting selezionato.

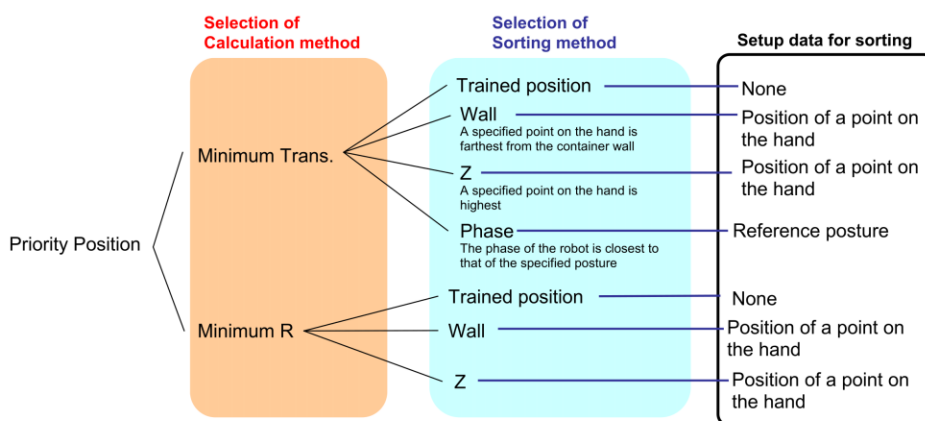


Figura 2.11: Metodo di calcolo e metodo di selezione delle posture calcolate con l'Interference Avoidance [4].

- X, Y, W, P, R : questi parametri sono identificati come *avoidance range*. Andando infatti a selezionare la singola coordinata mediante il box 'Enable', è possibile impostare l'intervallo di valori, rispetto al Tool frame, nei quali la nuova posizione ed orientazione è calcolata. In Fig. 2.12 a,b,c sono illustrati rispettivamente i range nelle coordinate X-Y, nella W e P ed infine nella R.

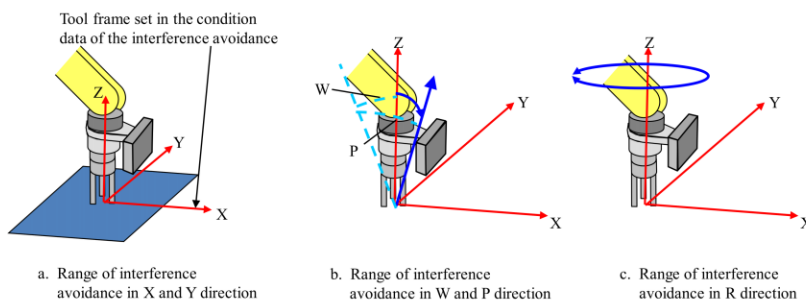


Figura 2.12: Rappresentazione degli avoidance range [4].

2.4.3 Programmazione: Vision e Teaching Pendant

Come ampiamente discusso, per l'applicazione di bin picking è necessario, oltre all'appena descritte impostazioni per evitare le collisioni, che vi sia un processo atto al riconoscimento effettivo dei componenti da movimentare. Questo è chiamato *Vision Process Tool* o *Vision Process* e si occupa di eseguire il SEARCH ovvero il rilevamento di componenti all'interno di un contenitore, a seguito di un'acquisizione eseguita con il 3D Area Sensor. Tale processo fornisce in uscita posizione ed orientazione dei pezzi riconosciuti, che dovranno poi essere inviate al robot in modo che vada ad eseguire la presa di questi, seguendo una logica

scelta dal programmatore in base all'applicazione specifica, e nel rispetto dell'Interference Avoidance. Il programma completo per la gestione dell'intero processo nel sistema Fanuc è chiamato *TP Program*.

Vision Process Tools

Nel sistema Fanuc è possibile creare un Vision Process tramite la finestra *iRVision Setup* nella Robot Homepage, se si utilizza come in questo caso un sistema 3DA, è opportuno utilizzare un 3D Area Sensor Vision Process. Questo specifico processo mette a disposizione numerosi *Vision Tools*, ossia specifiche funzioni software. Un Vision Process è infatti composto da una serie di Vision Tools, che verranno poi eseguiti in successione secondo l'ordine in cui sono disposti nel Vision Process stesso. Quest'ultimo necessita dei parametri di configurazione, i principali sono il *Camera Data* (ossia il file di calibrazione), lo User frame da utilizzare per i calcoli ed i 'Measurement' per definire il punteggio *Pick Score* secondo il quale saranno ordinati i componenti riconosciuti (di fatto qui viene definita la priorità, ad esempio è possibile dare priorità ai componenti più in alto nel contenitore, scegliendo Normalized Z come Measurement). Solitamente il primo tool è uno Snap Tool, atto ad acquisire un'immagine 2D per telecamera, nel quale è possibile anche ridurre la finestra acquisita, alleggerendo il carico computazionale. Segue poi un tool chiamato *3D Data Preprocess Tool*, i principali parametri in esso modificabili sono lo spessore del fondo del container, la forma dello stesso (in modo che i componenti vengano cercati solo al suo interno) e la sua altezza. I Vision Tools disponibili nel 3D Area Sensor Vision Process sono illustrati in Fig 2.13. In questo documento vengono approfonditi due Vision

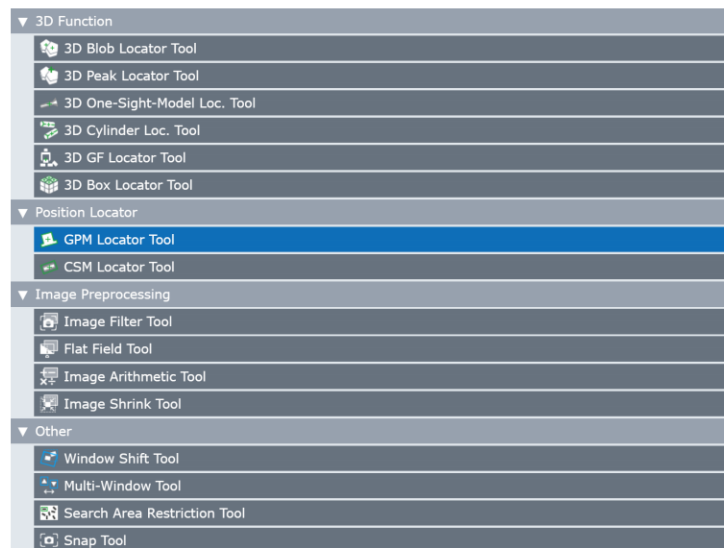


Figura 2.13: Finestra di setup per la scelta dei Vision Tools disponibili nel 3D Area Sensor Vision Process.

Tools, sono i più adatti per il riconoscimento di oggetti casualmente disposti e le loro performance verranno poi confrontate nel Cap³.

- **3D One-Sight-Model Locator Tool:** fa parte del gruppo delle funzioni 3D ed è uno dei tool più potenti e recenti messi a disposizione da Fanuc. Questo tool crea un modello 3D a partire da un'acquisizione o da un modello in formato *.STL* del componente, utilizza poi tale modello per rilevare lo stesso nella mappa 3D acquisita dal sistema di visione ed ottenere posizione ed orientazione del componente riconosciuto. Il termine 'One Sight', letteralmente 'una vista', indica il fatto che il modello impostato è relativo solo ad una vista del componente, ad esempio la vista frontale. Se un componente presenta viste differenti perché asimmetrico ed è richiesto di riconoscerlo da tutte le angolazioni, è necessario utilizzare più tools, aventi diverse viste del modello. Tale tool richiede in input una mappa 3D, ottenuta mediante un precedente 3D Data Preprocess Tool o un altro One-Sight-Model Locator Tool, il modello con cui andrà a fare il matching 3D come detto può derivare da un'acquisizione del sistema di visione o da un modello 3D. Nel primo caso è sufficiente porre un componente nel FOV del sistema di visione, con la faccia che si vuole utilizzare poi per il matching visibile dallo stesso. Selezionando poi la voce 'Area' nella sezione Model Train dei parametri, si apre quindi una finestra nella quale è visualizzata un'acquisizione, è sufficiente racchiudere all'interno di una selezione rettangolare il componente precedentemente citato (Fig^{2.14}). In questo modo e mediante successive maschere manualmente modificabili si ottiene il modello per il matching, di fatto è una porzione del pointcloud ottenuto da un'acquisizione. Per utilizzare il metodo con modello in formato *.STL* è necessario selezionare 'Import File' nella stessa sezione Model Train, la procedura prevede di inserire un modello, importandolo da un supporto di

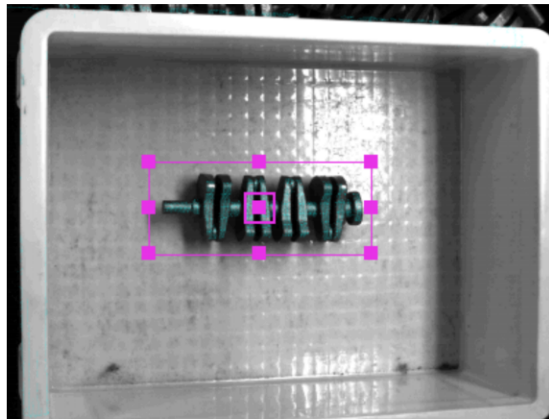


Figura 2.14: Finestra di selezione del modello in Model Train nel caso di modello ottenuto da mappa 3D ⁵.

memoria collegato al controllore (ad esempio una memoria USB collegata alla porta USB del controllore stesso). Di questo l'utente sceglie una vista, X,Y o Z sulla base del sistema di riferimento del modello stesso. Successivamente è possibile mascherare le parti della vista del componente che si ritengono non utili al riconoscimento ed infine selezionare il numero di punti, sulla superficie della vista selezionata, utili al riconoscimento.

- **GPM Locator Tool:** questo tool fa uso dello strumento di elaborazione delle immagini che fa parte delle funzioni più importanti di *iRVision*. Dopo aver insegnato a tale tool un model pattern, andrà a cercare tale pattern in un'immagine catturata dalle fotocamere e fornirà in output la posizione rilevata di tale modello. Il model pattern è insegnato al tool mediante una selezione rettangolare di un'acquisizione, similmente a quanto visto in precedenza per il One Sight con modello da mappa 3D. Il tool verifica la presenza di tale model pattern nell'immagine anche sulla base di 'parametri di elasticità', che permettono di rilevare il modello anche se differisce leggermente da quello impostato, ad esempio se è ruotato, inclinato o scalato. Tuttavia questo tool fornisce solo informazioni sulla posizione X-Y del modello rilevato, non sulla sua coordinata Z, per questo è di default associato un tool figlio chiamato *3D Plane Measurement Tool*. Quest'ultimo ha in input una mappa 3D, ad esempio un precedente 3D Data Preprocess Tool, e mediante un'ulteriore selezione all'interno di un pattern riconosciuto, è possibile definire la porzione di pointcloud utilizzato dal tool per il calcolo della coordinata di profondità Z.

Una volta conclusa la configurazione del Vision Process, al fine di poterlo gestire ed utilizzare in un programma nel robot, congiuntamente alle impostazioni di Interference Avoidance, è necessario configurare il Parts List Manager. Vi si accede dalla ROBOT Homepage, nella sezione *iRVision*, nella finestra che appare è possibile aggiungere o modificare Parts List. Una Parts List utilizzata in una cella dotata di sistema 3DA deve essere impostata di tipo SEARCH, accedendo poi alla Parts List si hanno a disposizione cinque diverse schermate di modifica:

1. **SEARCH VP List:** In questa pagina è possibile impostare il Vision Process da utilizzare, impostando anche quale Measurement dello stesso VP deve essere utilizzato per valutare la priorità dei componenti rilevati;
2. **PICK Position List:** in questa schermata (Fig. 2.15) è possibile impostare i parametri per il calcolo della posizione di PICK quindi il processo di visione da utilizzare, l'ID del modello da cercare in quello specifico VP e tutte le impostazioni relative all'Interference Avoidance. Nello specifico, si seleziona l'Interference Avoidance Setup System, Robot e Condition, in questo modo l'Interference Avoidance è completamente definita. In aggiunta è possibile impostare in 'Tofs', un registro di posizione che il controllore

utilizzerà come offset per il calcolo della posizione di approach al componente. Ad esempio impostando nel registro di posizione 100 una postura del tipo (0,0,100,0,0,0) il robot eseguirà un approach al pezzo, fermandosi alla distanza di 100mm in Z, di fatto sopra il componente. Infine è possibile utilizzare una posizione di PICK di riferimento, la maggior parte delle applicazioni non la richiede, tali campi vengono quindi solitamente posti tutti pari a 0;

#	Comment	Vision Process Name	Model ID	Interference Setup	Approach Setup	Reference PICK Position
1		V_3DA_SAMPLE	1	(V_SYS,V_ROB,V_CND)	(V_CND, 0, 100)	Not Set
2		Not Set	Not Set	Not Set	Not Set	Not Set
3		Not Set	Not Set	Not Set	Not Set	Not Set
4		Not Set	Not Set	Not Set	Not Set	Not Set

Comment

Use Found Position Enable

Vision Process Name Not Trained Sample Vision Process for 3D Area Sensor

Model ID

Interference Setup

Calculate IA Enable

IASYS UF:2

IAROB

IACND UT:4

Approach Setup

IACND UT:4

Ofs PR Not Used

Tofs PR

Reference PICK Position

PICK Position X mm

PICK Position Y mm

PICK Position Z mm

PICK Position W °

PICK Position P °

PICK Position R °

[PAGE] SAVE END EDIT

Figura 2.15: Finestra di modifica della PICK Position List.

3. **Push Part Data Setup:** quando nella Parts List si esegue un push di un part data, è necessario eliminare i part data precedenti. In questa schermata è possibile impostare quanto devono essere precedenti i part data per essere eliminati. Il sistema rende disponibile anche una funzione di duplication check che si occupa di controllare che non vengano aggiunti alla Parts List dei part data che appartengono ad un componente già all'interno della stessa lista. Evita quindi che vi siano dei duplicati nella Parts List, un part data viene giudicato duplicato se simile ad uno già presente nella lista secondo un range di distanza in XY, di default è impostato a 10mm. Se un componente rilevato dal sistema ha delle coordinate simili, all'interno del range appena indicato, a quelle di un componente già presente nella Parts List, non viene aggiunto alla stessa;

4. **Status Setup List:** in una Parts List ad ogni part data, ossia ad ogni componente rilevato, viene assegnato uno stato, i principali sono:

- **AWAITING:** è lo stato assegnato ad un part data quando viene aggiunto (push) alla Parts List. Rappresenta quindi il fatto che tale part data sia in attesa del pop;
- **PICK SUCCESS:** l'operazione di presa dal contenitore è avvenuta con successo;
- **PICK FAIL:** l'operazione di presa dal contenitore non è andata a buon fine;
- **PICK IA FAIL:** il calcolo della posizione di presa non è andato a buon fine a causa dell'Interference Avoidance. Non è quindi stato possibile calcolare una posizione ed orientazione che evitasse le collisioni, il componente non può quindi essere preso;
- **PICK CL FAIL:** quando si verifica una collisione mentre il robot stava eseguendo l'operazione di presa del componente.

Quando un componente viene prelevato dal contenitore, può essere eliminato dalla Parts List. Quando invece non è possibile prendere un componente, per evitare che il robot tenti ciclicamente di prelevarlo, oltre ad associare lo stato del part data a PICK FAIL, lo stesso part data viene aggiunto ad una Black List. Qualsiasi tipo di FAIL rilevato per un part data prevede che esso sia aggiunto a tale Black List. Mediante questi stati, i part data presenti nella part list, associati ai componenti nel contenitore, sono aggiornati costantemente, affinché abbiano lo stesso stato dei reali componenti che rappresentano. Nella schermata di Push Part Data Setup è possibile impostare la logica di aggiunta alla Black List e di eliminazione dei part data.

5. **Part Data Monitor:** è una schermata utile per controllare tutte le informazioni riguardanti la Parts List, come ad esempio il numero di push e pop eseguiti, quanti elementi sono presenti nella lista ed il vision process utilizzato.

TP Programming

Al fine di far eseguire al robot uno specifico task, come può essere quello del bin picking descritto nel presente documento, è necessario creare un programma, cui ci si riferisce solitamente come *TP Program*, nel quale sono raccolte le istruzioni che specificano al robot cosa deve fare. Nello specifico, per un'applicazione di bin picking con il 3D Area Sensor, il programma deve seguire il diagramma di flusso di Fig. [2.16](#)

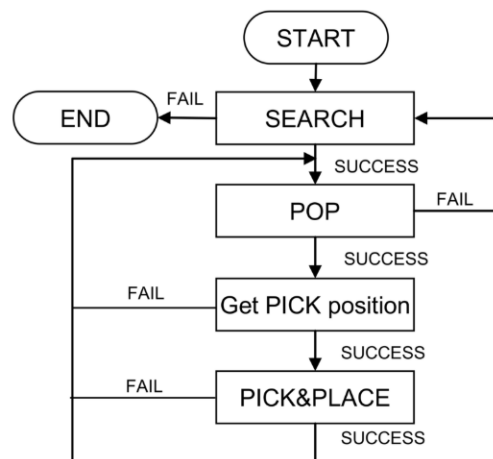


Figura 2.16: Flow chart di un programma per bin picking nel sistema Fanuc con 3DA [4].

Di seguito è riportato il programma di esempio *V_3DA_BP.TP* presente di default nel controllore ed utile per descrivere le fasi del processo e le singole funzioni macro che richiamate.

```

1  UFRAME_NUM=1
2  UTOOL_NUM=1
3  CALL BINPICK_CLEAR("Parts List ID"=1)
4
5  !SEARCH
6  LBL[1]
7  L P[1:Search] 2000mm/sec FINE
8  CALL BINPICK_ACQUIRE3DMAP("3D Area Sensor"='V_3DA_CAM')
9  CALL BINPICK_SEARCH("Parts List ID"=1,"SEARCH VP ID"=1,"Status R"=10)
10 IF R[10:Search Status]<>0,JMP LBL[999]
11
12 !POP
13 LBL[2]
14 CALL BINPICK_POP("Parts List ID"=1,"Status R"=11,"Model ID R"=12)
15 IF R[11:Pop Status]<>0,JMP LBL[1]
16
17 !GETPICKPOS
18 CALL BINPICK_GETPICKPOS("Parts List ID"=1,"PICK Pos ID"=1,"Status R"=13,
19 "PICK Pos PR"=20,"IA Offset PR"=21,"APP Pos PR"=22)
20 IF R[13:Pick Status]=0,JMP LBL[3]
21
22 CALL BINPICK_SETSTAT("Parts List ID"=1,"Part Status"=22)
23 JMP LBL[2]
24
25 !PICK
26 LBL[3]
27 L P[2:Pick Approach] 2000mm/sec CNT100
28 L PR[22:Pick App.] 2000mm/sec CNT50
29 L PR[20:Pick] 500mm/sec FINE
30 !Insert program instructions to grasp the part.
31 CALL ...
32 CALL BINPICK_SETSTAT("Parts List ID"=1,"Part Status"=20)
33 L PR[22:Pick App.] 500mm/sec CNT50
  
```

```

34 L P[2:Pick Approach] 2000mm/sec CNT100
35
36 !PLACE
37 L P[3:Place Approach] 2000mm/sec CNT100
38 L P[4:Place] 500mm/sec FINE
39 !Insert program instructions to place the part.
40 CALL ...
41 L P[3:Place Approach] 2000mm/sec CNT100
42
43 !CONTINUOUS
44 JMP LBL[2]
45
46 !END
47 LBL[999]
48 J P[5:Home] 100% FINE

```

Prima di descrivere nel dettaglio il programma è opportuno definire in anticipo le seguenti posizioni:

P[1: Search]	Posizione con il robot fuori dal FOV del sistema di visione, qui attende che venga eseguito il SEARCH.
P[2: Pick Approach]	Posizione con end effector sopra il contenitore.
P[3: Place Approach]	Posizione con end effector sopra la posizione dove verrà poggiato il componente.
P[4: Place]	Posizione dove verrà poggiato il componente.
P[5: Home]	Posizione dove il robot si porta ad inizio e fine programma, può essere la stessa di P[1].

Il programma V_3DA_BP.TP si compone come segue:

- Riga 1-2: vengono definiti i frame utilizzati nel programma, lo User ed il Tool rispettivamente;
- Riga 3: viene chiamata la funzione macro *BINPICK_CLEAR()*, che elimina tutti i part data nella Parts List specificata nell'argomento;
- Riga 5÷10: in questo blocco si eseguono le operazioni necessarie per l'operazione di SEARCH. L'indicazione *LBL* rappresenta un'etichetta, da richiamare quando il programma è in una diversa riga e si vuole saltare a quella specifica riga, utilizzando l'operatore *JMP* ossia jump. In riga 7 il robot viene portato, con un movimento di tipo 'linear' e con velocità 2000mm/s, alla posizione 1 definita Home. Successivamente viene chiamato il programma *BINPICK_ACQUIRE3DMAP()*, questo esegue un'acquisizione attraverso il 3D Area Sensor specificato nel file di calibrazione che è impostato come parametro. Alla riga 9 viene richiamato il programma *BINPICK_SEARCH()*, è la macro che permette di aggiungere part data

alla Parts List. Richiede tre argomenti nel seguente ordine: il numero della Parts List alla quale si vogliono aggiungere i part data, il numero del processo di visione per il SEARCH all'interno della SEARCH VP LIST ed infine il numero identificativo del registro che rappresenterà lo stato del processo di SEARCH stesso. Questa funzione inserisce in tale registro il valore 0 se sono state aggiunte part data alla Parts List o il valore 1 se non ne sono state aggiunte. Nell'esempio il registro è il numero 10, se dopo la chiamata di `BINPICK_SEARCH()` contiene un valore diverso da 0, allora non vi sono part data nella lista, quindi il programma passa alla label 999 ossia riga 47, dove si riporta il robot alla posizione di Home ed il programma termina;

- Riga 12÷15: in questo blocco si eseguono le operazioni necessarie per l'operazione di POP, quindi per estrarre dalla Parts List il part data con priorità maggiore. A tal fine viene richiamata la funzione `BINPICK_POP()`, che richiede i seguenti argomenti: il numero della Parts List, il numero identificativo del registro rappresentativo dello stato dell'operazione di POP (0=Success, 1=Fail, nessun part data può essere estratto) ed il numero identificativo del registro in cui viene salvato il model ID del part data estratto. Se tale POP status è diverso da 0, nessun part data è stato estratto, quindi il programma passa a label 1 eseguendo nuovamente un SEARCH;
- Riga 17÷20: in questo blocco si eseguono le operazioni necessarie per ottenere la posizione di presa del componente, sulla base del part data estratto in precedenza. A riga 18 viene per questo richiamata la funzione `BINPICK_GETPICKPOS()` che necessita dei seguenti argomenti:
 - numero della Parts List;
 - numero della posizione di PICK all'interno della PICK POSITION LIST;
 - numero identificativo di un registro che rappresenterà lo status dell'operazione di calcolo della posizione di PICK (0=Success, 12=Fail impossibile ottenere una posizione di PICK che soddisfi l'Interference Avoidance, 13=Fail impossibile ottenere una posizione di approach che soddisfi l'Interference Avoidance);
 - numero identificativo del registro di posizione in cui salvare la posizione di PICK calcolata. Nell'esempio è il registro di posizione 20;
 - numero identificativo del registro di posizione in cui salvare il tool offset calcolato dall'Interference Avoidance, è applicato alla posizione di PICK. Nell'esempio è il registro di posizione 22;
 - numero identificativo del registro di posizione in cui salvare la posizione di Approach calcolata.

Se lo status del calcolo è pari a 0, il programma prosegue alla label 3, dove avvengono le operazioni per eseguire la presa del componente;

- Riga 22-23: viene richiamata la funzione *BINPICK_SETSTAT()*, che richiede due argomenti, il numero della Parts List e lo status da associare al part data estratto in precedenza. Tale status è codificato come segue: 20=PICK Success, 21=PICK Fail, 22=PICK IA Fail, 23=PICK CL Fail. In questo caso, segue il Fail del calcolo della posizione di PICK a causa dell'Interference Avoidance quindi lo status viene aggiornato a 22 ed il programma si riporta alla label 2 in modo da poter eseguire il POP di un altro part data;
- Riga 25÷34: in questo blocco si eseguono le operazioni necessarie per eseguire la presa del componente. Viene quindi portato il robot alla posizione P2 di Pick Approach, poi viene eseguito l'approach al componente, portando il robot alla posizione salvata in precedenza in PR22, ed in seguito alla PR20 ossia alla posizione effettiva del componente. In riga 31 viene richiamato il programma macro per prendere il componente, ad esempio che esegua la chiusura del gripper, in seguito lo status del part data del componente viene aggiornato a 20 ossia PICK Success. Viene poi riportato il robot alla posizione di approach, per poi portarsi nuovamente alla posizione sopra il contenitore P2;
- Riga 36÷41: in questo blocco si eseguono le operazioni necessarie per eseguire il place del componente. A tal fine il robot viene portato prima alla posizione P3 di approach al place poi alla effettiva posizione P4 di place. Segue il richiamo della funzione che agisce sull'end effector per rilasciare il componente (ad esempio apre le griffe della pinza) e infine il robot viene riportato alla posizione P3;
- Riga 43-44: il programma è ciclico, quindi a seguito del place di un componente, torna alla label 2 quindi a fare il pop del part data successivo nella Parts List. Nel caso non vi fossero più componenti ossia part data nella lista, il programma passa alla label 1 eseguendo una nuova scansione 3D. Se non vengono rilevati componenti il programma passa alla label 999, dove il robot viene riportato alla posizione P1 di Home ed il programma stesso termina.

Capitolo 3

Riconoscimento di oggetti con geometrie complesse

3.1 Introduzione

Come descritto nel Cap [2.3](#) il sistema di visione impiegato nella cella analizzata nel medesimo capitolo è specificatamente studiato per la depallettizzazione ed il bin picking industriale. Tuttavia, in riferimento a quest'ultimo, nella manualistica è specificato che gli oggetti da riconoscere non devono essere trasparenti e devono essere di materiale meno riflettente possibile. In aggiunta, anche le dimensioni dei componenti stessi sono una caratteristica fondamentale per il corretto funzionamento del sistema. Il depallettizing è una delle attività per la quale il sistema è più adatto, come anticipato, quindi gli oggetti idonei al riconoscimento sono scatole, che hanno dimensioni variabili ma essendo caricate su europallet (800x1200mm) solitamente non hanno dimensioni inferiori a 400x400x300mm. Per questo motivo il sistema in esame risulta teoricamente



Figura 3.1: Prelievo di stampi metallici da un bidone per la sbavatura.

poco adatto al riconoscimento di oggetti di dimensioni inferiori e con geometrie più complesse del semplice parallelepipedo rappresentato dalla scatola. La stessa manualistica Fanuc, per il bin picking, fa riferimento ad applicazioni dove i componenti sono di forme molto semplici, ad esempio stampi metallici non riflettenti come in Fig. 3.1 o di dimensioni considerevoli come i già citati alberi a gomiti di Fig. 2.14. Il seguente capitolo si propone quindi di analizzare e testare le performance del Fanuc 3D Area Sensor per valutarne i limiti in termini di complessità della geometria degli oggetti riconosciuti e delle caratteristiche del materiale degli stessi.

3.2 Test sistema di visione e Vision Tools

Al fine di valutare le performance del sistema sono stati eseguiti diversi test, che hanno previsto l'utilizzo di un oggetto specifico come componente da riconoscere e di entrambi i Vision Tools descritti nel Cap. 2.4.3 ossia il GPM Locator Tool ed il 3D One-Sight-Model Locator Tool. In particolare, quest'ultimo è stato utilizzato con entrambe le modalità di teaching del modello, quindi da importazione di file .STL e da acquisizione dell'oggetto stesso. Le principali metriche per valutare le prestazioni sono il numero di oggetti riconosciuti nell'acquisizione, la qualità del loro riconoscimento ed il tempo necessario per ottenerlo. Tutti i test sono stati eseguiti con i componenti disposti all'interno di un contenitore 300x400x150mm. L'oggetto con il quale è stato fatto il confronto tra i Vision Tools è un dado M30 in acciaio (Fig. 3.2), che per dimensioni, forma e tipo di materiale risulta essere un ottimo candidato, viste le problematiche citate in precedenza.

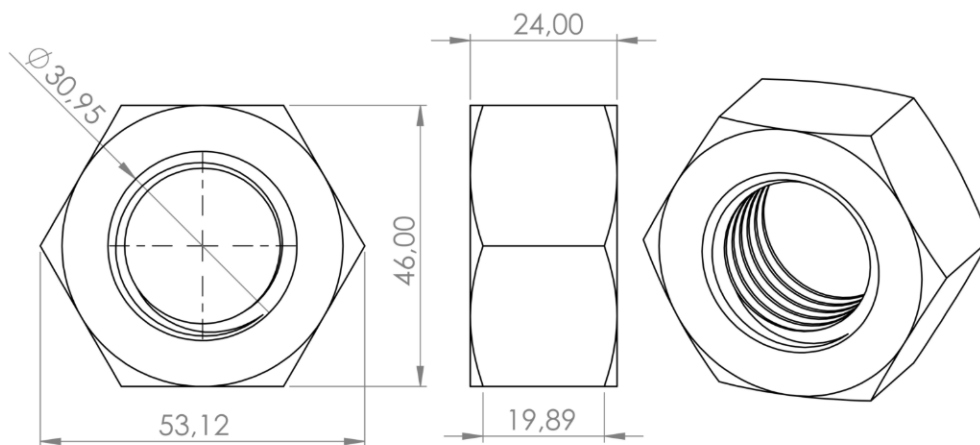


Figura 3.2: Principali dimensioni dado M30 ISO.

3.2.1 GPM Locator Tool

Come descritto nel Cap. 2.4.3 questo tool funziona grazie ad un riconoscimento 2D di un modello, successivamente vengono considerati i punti della mappa 3D appartenenti al modello appena riconosciuto, in modo da poter ottenere l'informazione relativa alla coordinata Z dello stesso. Per impostare il modello che poi il processo cercherà nell'acquisizione, è stato posto un singolo componente al centro del contenitore ed è stata effettuata un'acquisizione. Da questa poi, mediante una selezione rettangolare, è stato selezionato il componente all'interno dell'acquisizione. A tale porzione di acquisizione è stata poi applicata una maschera in modo da eliminare le caratteristiche non appartenenti al dado, infine la geometria dello stesso è stata migliorata mediante l'applicazione di forme standard (cerchi e linee spezzate), il tutto utilizzando la già citata interfaccia di *iR*Vision della ROBOT Homepage (Fig. 3.3a). Il modello che poi il controllore andrà a ricercare successivamente nelle acquisizioni è illustrato in Fig. 3.3b. Una volta completato il settaggio del Vision Process, sono stati eseguiti test per

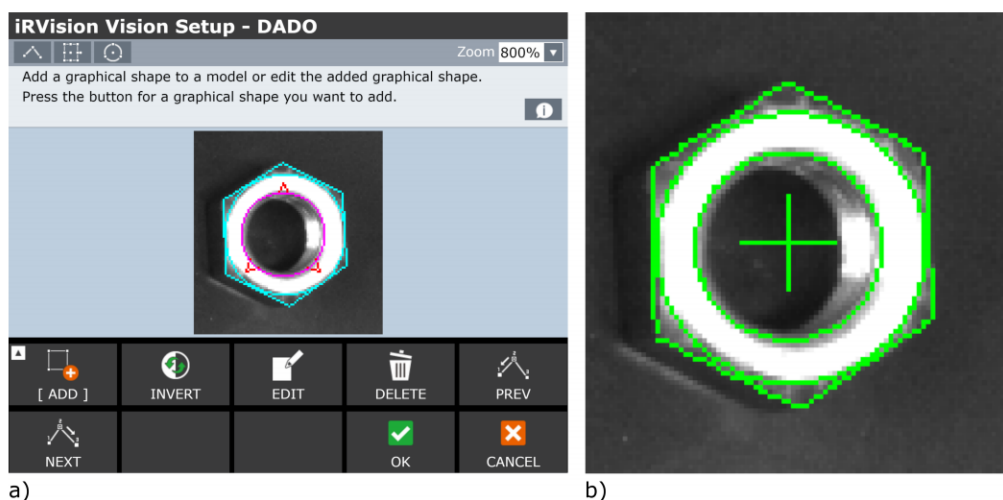


Figura 3.3: a) Interfaccia di *iR*Vision per il teaching del model pattern nel GPM Locator Tool, b) model pattern del dado ottenuto.

il riconoscimento del componente appena descritto, disponendo diversi dadi nel contenitore, in configurazioni via via più complesse. La prima configurazione, quindi la più semplice, ha previsto nove dadi disposti ordinatamente in fila, con alcuni aventi la stessa configurazione del model pattern, ed altri semplicemente ruotati di pochi gradi rispetto allo stesso modello di riferimento, come in Fig. 3.4a. L'intero processo di visione, denominato in questo caso 'DADO', è un 3D Area Sensor Vision Process, composto da uno Snap Tool, un 3D Data Preprocess Tool, un GPM Locator, con un 3D Plane Measurement Tool come figlio, ed infine un Offset data Calculation Tool, come illustrato in Fig. 3.4b. Nello specifico, in Fig. 3.5a,b sono illustrati i parametri dei due Vision Tools analizzati, ossia il

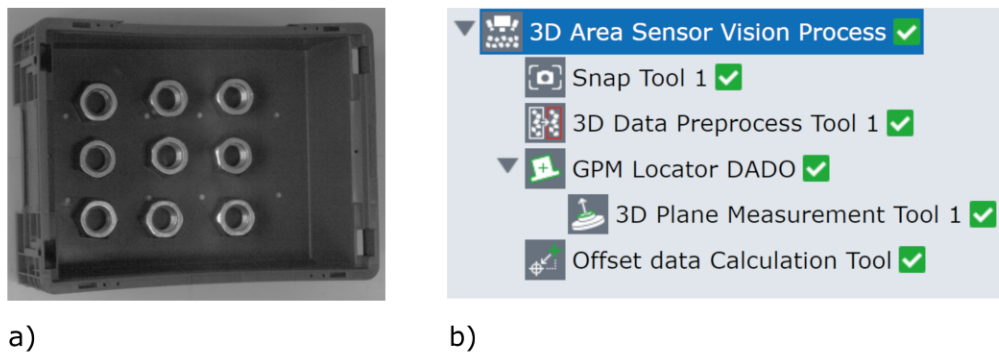


Figura 3.4: a) prima configurazione di test, b) struttura ad albero del Vision Process utilizzato per il riconoscimento mediante GPM dei dadi.

GPM Locator ed il 3D Plane Measurement rispettivamente. Questi parametri sono rimasti invariati durante i test con le successive configurazioni degli oggetti. Il GPM Locator Tool mette a disposizione un primo blocco di impostazione, relativo al teaching del model pattern, nel quale sono state eseguite le operazioni

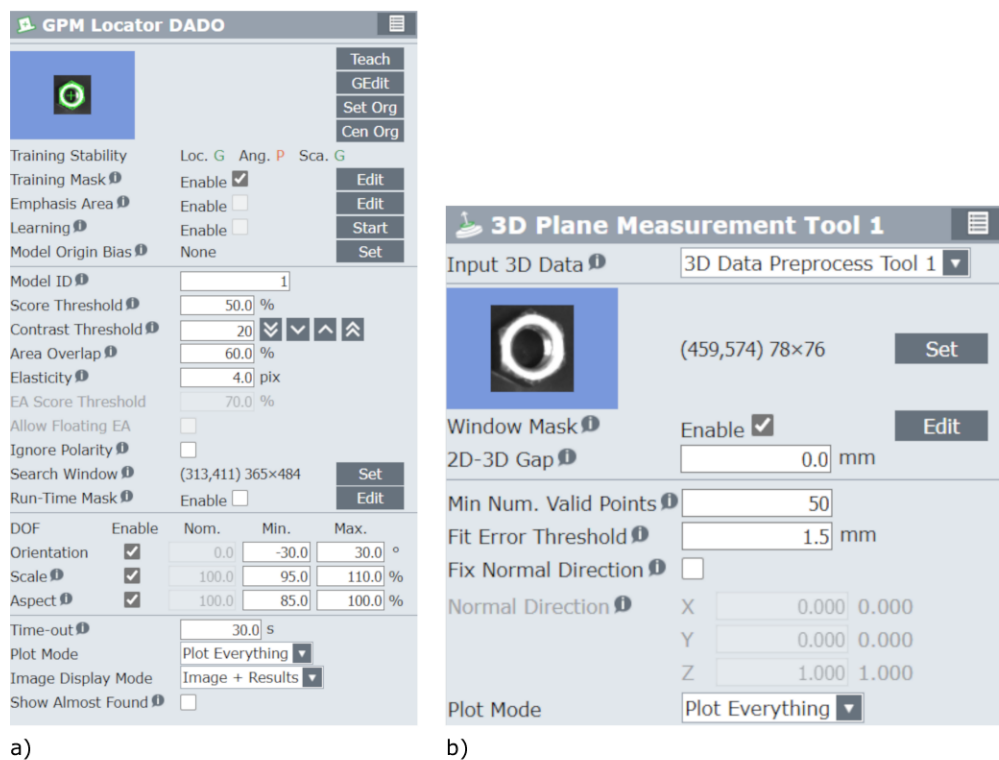


Figura 3.5: a) interfaccia di modifica dei parametri del GPM Locator Tool, b) interfaccia di modifica dei parametri del 3D Plane Measurement Tool.

descritte in precedenza. Il secondo e terzo insieme di settaggi sono relativi al riconoscimento vero e proprio del modello, in particolare si ha la soglia di punteggio di riconoscimento al disotto della quale un oggetto non viene riconosciuto, la soglia di contrasto, l'Area Overlap (se viene riconosciuto lo stesso pattern più volte sullo stesso oggetto, viene considerato solo quello con punteggio maggiore) e l'elasticità. Quest'ultimo è un importante parametro perché, insieme ai successivi *Orientation*, *Scale* ed *Aspect*, permette di definire quanto può differire il componente rispetto al model pattern, affinché il primo venga riconosciuto anche se disposto diversamente rispetto al modello di riferimento. Il parametro *Orientation* permette di definire il range angolare nel quale il model pattern può essere ruotato, lo *Scale* riguarda invece la dimensione dello stesso modello, che è necessario far variare in quanto gli oggetti non sono tutti alla stessa distanza dal sistema di visione, per questo risultano più grandi o più piccoli a seconda della loro quota. *Scale* invece permette di aggiungere un'elasticità di forma al model pattern, utile in quanto gli oggetti che si trovano nella periferia del campo di visione delle telecamere tendono ad essere leggermente distorti a causa delle lenti presenti sulle telecamere stesse. Il tuning dei parametri appena citati è di fondamentale importanza, perché sono ciò che definiscono la funzionalità del GPM Locator di trovare il model pattern nelle immagini catturate dal sistema. La loro scelta e definizione è basata su numerose prove eseguite, nella configurazione illustrata ed in quelle successive, saranno discusse in seguito le criticità. Il 3D Plane Measurement Tool trova un piano in una certa zona della mappa 3D, essendo figlio del GPM Locator, va a calcolare il piano all'interno dell'area individuata dal model pattern. Ricapitolando, il GPM rileva i dadi nell'immagine 2D, il risultato è rappresentato dai quadrati verdi e relativi model pattern del dado in Fig.3.6, successivamente il Plane Measurement calcola il piano all'interno dei model pattern individuati, rappresentati dai quadrati blu all'interno dei quadrati verdi nella medesima immagine. Il piano è come detto calcolato sulla base dei punti 3D acquisiti che appartengono alla regione di spazio individuata dal model pattern del dado, sempre in Fig.3.6 è possibile vedere tali punti, che risultano distribuiti sulla superficie piana dei dadi. Per questo motivo tale piano coincide con quello che è possibile immaginare poggiato sulla superficie a vista del dado stesso. In questo modo si ottiene l'informazione in coordinate X,Y,Z e le rotazioni W,P,R delle superfici dei dadi, informazioni utili per individuarli nello spazio e per permettere al robot di eseguire la loro presa.

L'acquisizione mediante il sistema di visione impiega un tempo nell'intorno dei due secondi, costante, e non è possibile modificarlo. Il tempo di riconoscimento degli oggetti invece è, come detto, un parametro utile per stimare le prestazioni del tool e del sistema nel suo complesso. Tale tempo, nella configurazione analizzata, varia tra i 700 ed gli 800ms ed il risultato di un'esecuzione del relativo Vision Process appena discusso è illustrata in Fig.3.6. Si noti come tutti i dadi presenti nel contenitore siano stati riconosciuti correttamente, il model pattern risulta infatti perfettamente aderente su ogni dado nell'acquisizione, giustifi-

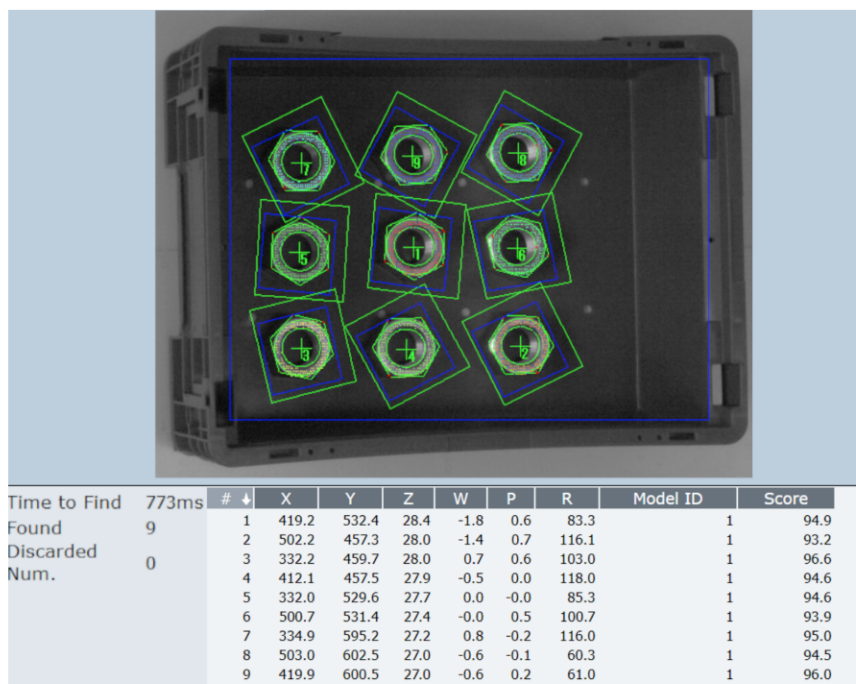


Figura 3.6: Risultato del Vision Process DADO: in basso a sinistra il tempo impiegato per la sua esecuzione, con il numero degli elementi trovati. In basso a destra una lista che rappresenta le coordinate nello spazio del centro (croce verde) di tali componenti riconosciuti, insieme al loro numero identificativo ed il punteggio, riferito al model pattern di riferimento.

quando gli elevati punteggi, espressi in centesimi, mai inferiori a 93. I numeri posizionati su ogni dado riconosciuto corrispondono all'ordine di priorità, che nel Vision Process in esame è stata definita come la quota Z. Si ha quindi il dado con numero 1 che è più in alto rispetto a tutti gli altri, logica che viene solitamente applicata nel bin picking visto che si predilige il pick dei componenti più in alto nel contenitore, visto che si tratta di svuotare lo stesso.

La seconda configurazione analizzata ha previsto, come anticipato, una maggiore complessità nella disposizione dei componenti, rappresentata da dadi inclinati e che si sovrappongono ad altri, come illustrato in Fig 3.7a. Il risultato del Vision Process DADO ha portato al risultato illustrato in Fig 3.7b, nel quale è possibile notare che sono stati rilevati 6 dadi dei 9 totali. In aggiunta, i componenti coperti parzialmente da altri sono stati riconosciuti con poca fedeltà, il loro punteggio è nell'intorno dei 70/100. Questa specifica configurazione consente di analizzare il principale problema dei sistemi di visione 3D a luce strutturata, rappresentata dagli oggetti di materiale riflettente. Come si può chiaramente notare nell'immagine con i risultati del processo di visione, gli unici componenti che non sono stati riconosciuti sono i dadi inclinati. Su tali componenti, la luce

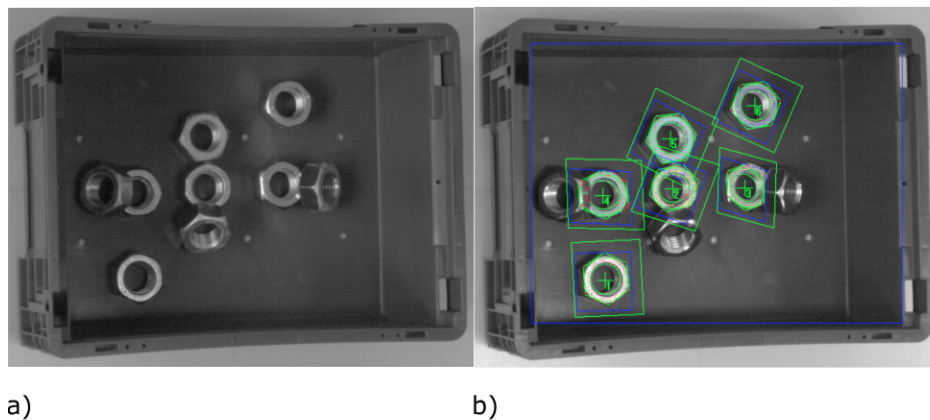


Figura 3.7: a) seconda configurazione analizzata, b) risultato del Vision Process DADO sulla seconda configurazione.

generata dal proiettore, viene riflessa, il sistema di visione, come ampiamente discusso nel Cap [1.2.3](#) si basa sulla rilevazione della modifica di tale pattern quando colpisce la scena. Nel caso di oggetti riflettenti, la luce viene deviata, non rendendo possibile al sistema la rilevazione di qualsiasi tipo di informazione riguardo alla zona riflettente. Questa mancanza di informazioni è chiaramente visualizzata in Fig [3.8](#) in questa immagine sono stati disposti due dadi con una leggera inclinazione, il dado di destra è come quelli utilizzati nelle configurazioni finora descritte, su quello di sinistra invece è stato applicato del nastro trasparente con finitura opaca direttamente sulla superficie del componente stesso, così da poter confrontare il caso di superficie riflettente e non. Grazie al 3D Data Preprocess Tool è possibile visualizzare i punti che compongono la mappa 3D. Di questi, quelli rossi sono filtrati dallo stesso tool perché appartenenti al conte-



Figura 3.8: Punti della mappa 3D risultati dal 3D Data Preprocess Tool su due dadi inclinati, il dado di destra è come quelli analizzati, quello di sinistra ha la superficie opaca grazie all'applicazione di nastro trasparente con finitura opaca.

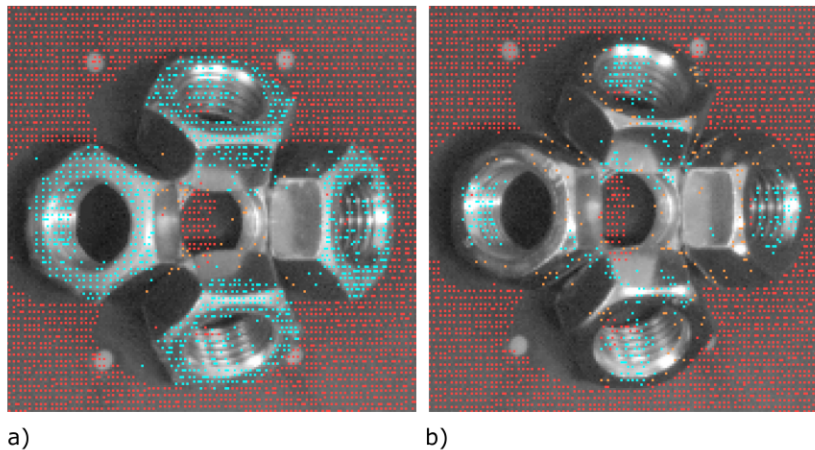


Figura 3.9: a) punti della mappa 3D su quattro dadi inclinati con superficie opaca, b) punti della mappa 3D su quattro dadi normali, inclinati allo stesso modo.

nitore, quelli azzurri invece sono relativi ai componenti. Risulta estremamente evidente che il numero di punti sul dado semplice è nettamente inferiore, si tratta di un numero di punti che si è attestato attorno ad 80 in tutti i test eseguiti su questa configurazione. Gli unici punti rilevati dal sistema sono relativi alla filettatura interna del componente, ma, insieme al ridotto numero di punti, rendono impossibile al 3D Plane Measurement Tool il calcolo del piano quindi di fatto il riconoscimento del componente stesso. Il dado con superficie non riflettente invece ha un numero di punti nell'intorno dei 200, per la maggior parte appartengono alla superficie del dado stesso, quindi utili al 3D Plane per il calcolo. Un'altra configurazione finalizzata alla valutazione della riflettività è illustrata in Fig. 3.9 ed ha portato a risultati analoghi a quanto appena descritto.

Questo test ha quindi dimostrato che la riflettività dei dadi che presentano un'inclinazione, come quelli nella seconda configurazione analizzata, non permette al sistema di visione di avere sufficienti informazioni sui componenti stessi, portando il Vision Process a non riconoscerli. Se si sostituiscono i dadi inclinati con gli stessi cui è stato applicato il nastro trasparente opaco, mantenendo la stessa configurazione di Fig. 3.7a, il processo di visione fornisce il risultato di Fig. 3.10. Il risultato è sicuramente migliore del precedente, tuttavia sono stati riconosciuti 8 dei 9 dadi, in aggiunta i componenti inclinati (numero 1 e 2 nella medesima figura) sono stati riconosciuti comunque con un punteggio inferiore (intorno a 70) rispetto a quelli piani. Questo non perfetto riconoscimento del model pattern non dipende dalla quantità di punti sullo specifico componente, bensì dal GPM Locator, che si è rivelato un tool estremamente sensibile alle condizioni dell'ambiente. Sebbene i tempi per l'esecuzione dell'intero vision process siano estremamente ridotti, sempre tra i 700ms ed i 900ms, i risultati dello stesso

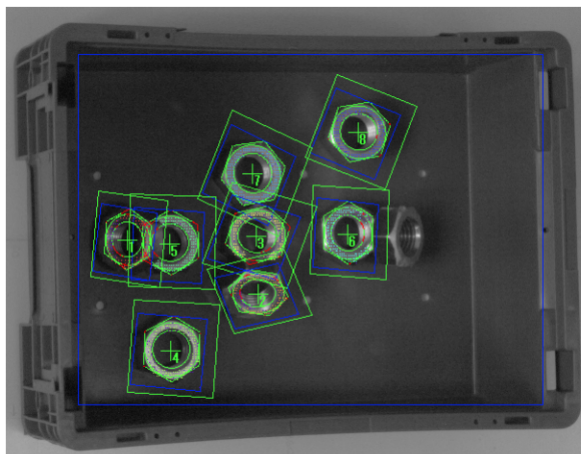


Figura 3.10: Risultato del Vision Process DADO sulla seconda configurazione analizzata, modificata con i dadi la cui superficie è stata resa opaca.

processo si sono rivelati estremamente variabili. Nel setup di laboratorio l'illuminazione non rimane costante durante l'intero arco della giornata, esattamente come nell'ambiente industriale, dove le condizioni di luce variano anche significativamente. Il Vision Process analizzato, avente il GPM Locator come tool per il riconoscimento degli oggetti, ha portato a risultati estremamente diversi se eseguito nella mattinata o nel pomeriggio. Al fine di ottenere risultati accettabili, quindi facendo in modo che vengano riconosciuti almeno i dadi più in alto, è stato necessario variare i parametri descritti in precedenza, più volte durante l'arco della giornata. Nello specifico, i parametri Contrast Threshold, Elasticity, Scale ed Aspect del GPM Locator hanno richiesto un tuning molto frequente, anche a distanze temporali molto ridotte (una volta ogni ora nella giornata di test più sfavorevole). Essendo parametri definiti solo grazie a numerosi test, dipendono infatti dalla configurazione e dalle condizioni dell'ambiente nello specifico momento del test stesso, non è possibile prevedere i loro valori, di conseguenza il GPM Locator risulta essere un tool inadatto all'applicazione di bin picking in ambienti con condizioni di illuminazione variabili. In aggiunta, il medesimo tool, anche se tarato grazie ad un test con i corretti valori dei parametri appena citati, porta spesso a risultati leggermente diversi in test successivi, eseguiti a qualche secondo di distanza. A titolo di esempio, in un test viene eseguita una scansione 3D, viene fatto eseguire il Vision Process e si ottiene il riconoscimento di 8 di 9 dadi, mantenendo invariata la configurazione, a distanza di 10s viene eseguita una nuova scansione con successiva esecuzione del processo di visione ed il risultato è il riconoscimento di 7 componenti dei 9 presenti. Oltre ad essere sensibile alle condizioni di illuminazione, il GPM si è quindi rivelato un tool estremamente instabile, fornendo come detto risultati variabili anche in condizioni di contorno costanti. In Fig. 3.11a,b sono illustrati i test su configurazioni più com-

plesse, la numero 3 e 4 rispettivamente, nelle quali il tool è riuscito a riconoscere solamente un componente, risultato insufficiente nel caso di un'applicazione di bin picking. Inoltre il componente riconosciuto nella configurazione 4 non è il componente più in alto, quindi il suo riconoscimento come componente numero 1 non è corretto. In conclusione, anche superando il problema intrinseco della riflettività che affligge i sistemi di visione a luce strutturata, mediante il GPM Locator Tool, supportato dal 3D Plane Measurement Tool, non si ottengono risultati sufficienti, in termini di riconoscimento degli oggetti, per poter impiegare tale coppia di tool in un processo di bin picking industriale. Per questo motivo è stato profondamente testato il già discusso 3D One-Sight-Model Locator Tool, trattato nella sezione seguente.

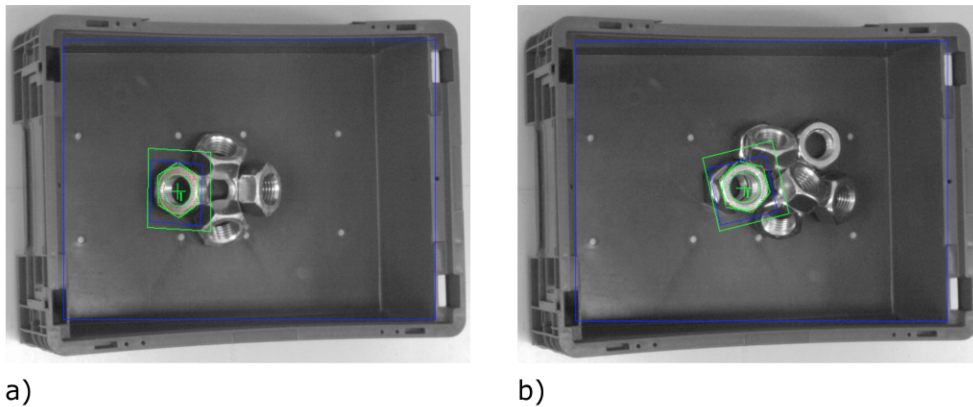


Figura 3.11: a) risultato Vision Process DADO su configurazione 3, b) risultato del medesimo Vision Process su configurazione 4.

3.2.2 3D One-Sight-Model Locator Tool

Questo Vision Tool, a differenza del precedente GPM che necessitava di un ulteriore tool figlio per ottenere le informazioni di un oggetto nello spazio, opera già nelle tre dimensioni. In input richiede una vista di un modello tridimensionale, che trasforma in un insieme di punti, il compito del tool è di trovare una o più istanze di tale modello all'interno della mappa tridimensionale acquisita mediante il sistema di visione. A tale procedura vi si riferisce spesso con il nome di *3D matching*, perché si tratta di far combaciare (match) il modello di riferimento con la forma tridimensionale del reale oggetto nella scena, che è composta di un insieme di punti. Affinché il tool non cerchi di fare il match del modello con punti che non potrebbero mai appartenere ad un oggetto, è consigliato eliminare i punti appartenenti al contenitore, in questo modo si ottiene anche il vantaggio di un tempo computazionale richiesto nettamente inferiore. Questa eliminazione dei punti appartenenti al contenitore o, in generale, alla scena senza oggetti, è resa possibile dal 3D Data Preprocess Tool, perché permette di definire lo spessore del fondo del contenitore, inteso come altezza tra il piano di lavoro ed il fondo della scatola stessa, in questo modo tutti i punti in una mappa 3D ad un'altezza inferiore non vengono considerati. Partendo da una mappa 3D come quella nell'esempio riportato in Fig. 3.12a, grazie a questa funzionalità si ottiene il risultato in Fig. 3.12b, dove i punti della mappa 3D non eliminati sono in azzurro ed appartengono proprio agli oggetti nel contenitore. Questa mappa 3D viene data come input al 3D One Sight Model Locator Tool, quindi questo andrà a fare il 3D matching solo con i punti dove vi è sicuramente un oggetto, come detto riducendo notevolmente l'onere computazionale. (Nella medesima immagine, si noti anche la mancanza di punti nel caso delle due lattine presenti nella parte destra del contenitore, questo per la stessa criticità dei materiali

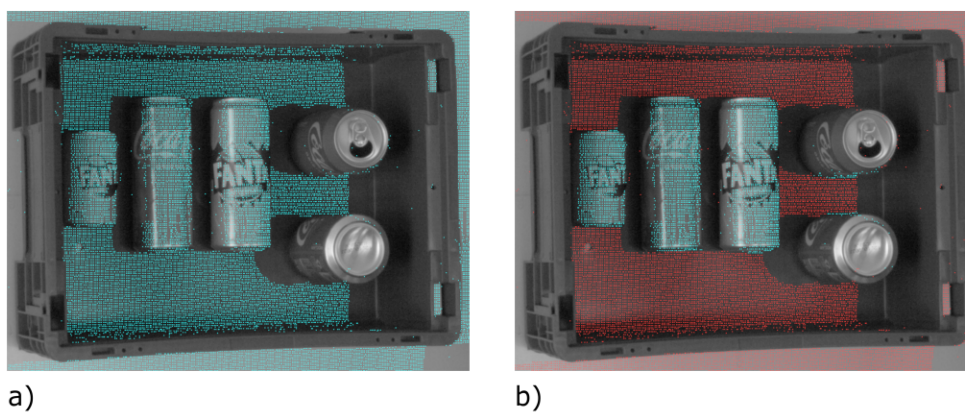


Figura 3.12: a) mappa 3D risultante dall'acquisizione di un contenitore con cinque lattine, b) stessa mappa 3D ma a valle dell'applicazione del filtro relativo all'altezza del fondo del container fornito dal 3D Data Preprocess Tool.

riflettenti descritta in precedenza, problema che non può essere risolto con nessun tool perché intrinseco di questo tipo di sistemi di visione). I test eseguiti per la valutazione delle performance del 3D One Sight Locator Tool sono stati eseguiti sui dadi disposti nelle medesime configurazioni illustrate in precedenza, in modo da poter ottenere un confronto diretto tra i due Vision Tools. Prima di descrivere i risultati ottenuti in quelle configurazioni, è opportuno descrivere i principali parametri di questo tool. Innanzitutto è necessario scegliere che tipo di modello utilizzare tra l'importazione di un modello in formato .STL e la selezione da mappa 3D, sono stati utilizzati entrambi in modo da poter valutare le performance di entrambi, di seguito viene descritta la procedura per il primo. Per importare un file in formato .STL è sufficiente caricare tale file su un supporto di memoria USB e collegarlo al controllore. Nella finestra di modifica del tool 3D One Sight Model, nella sezione 'Model Train' selezionare 'Import File' (Fig. 3.13a), sarà poi richiesto di inserire il percorso file del file stesso e la vista di tale modello che si vuole considerare. Nel caso del dado è stato importato un

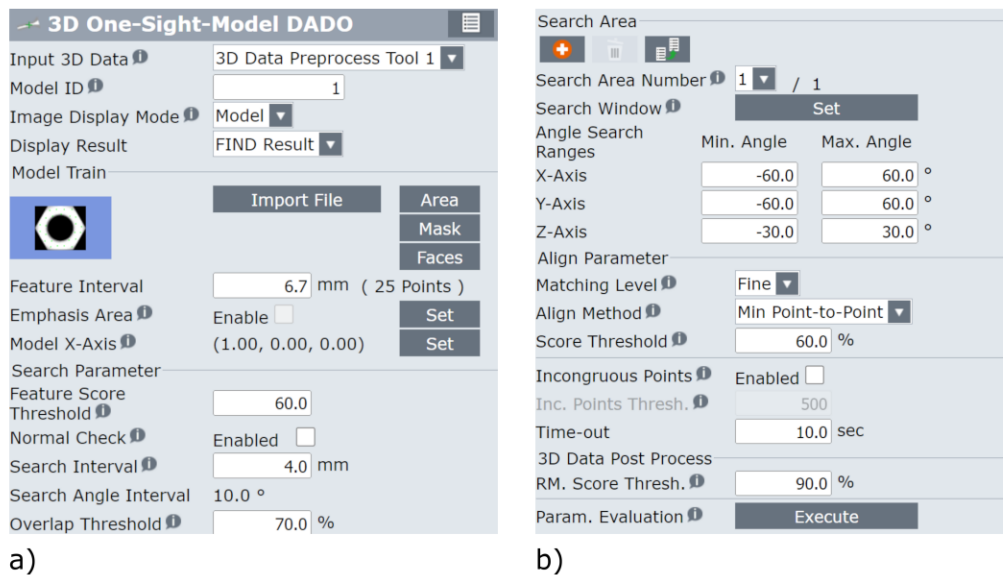


Figura 3.13: a,b) parametri del 3D One Sight Model Locator Tool.

file .STL ottenuto da un file CAD di un dado M30 con dimensioni reali, la vista selezionata è quella superiore, analoga a quella utilizzata dal GPM. La vista del componente selezionata viene trasformata dal processore in un insieme di punti, successivamente la procedura prevede di dividere in una o più aree la faccia selezionata e di eliminare eventuali punti che non sono utili al riconoscimento del componente. In questa applicazione è sufficiente una singola area e non sono stati rimossi punti dal modello importato, come illustrato in Fig. 3.14a. Successivamente la procedura prevede di definire il 'Feature Interval', che rappresenta quanti punti, di tale insieme appena definito, sono utili per il 3D matching. Si

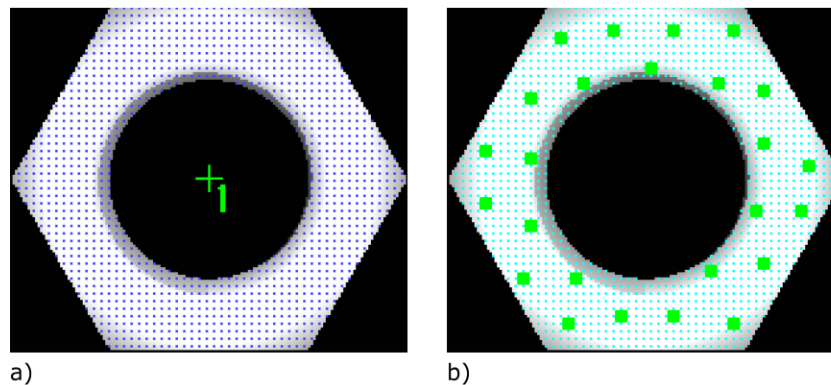


Figura 3.14: a) definizione della vista del modello, b) definizione del Feature Interval.

esprime in millimetri, un valore basso rende il matching estremamente preciso ma lento, al contrario un numero elevato richiede meno tempo per il riconoscimento degli oggetti ma è meno accurato. Nei primi test il valore è rimasto invariato a 6.7mm come consigliato di default (che corrisponde a 25 punti sul modello), successivamente è stato modificato per valutare un compromesso tra velocità ed accuratezza di riconoscimento dei dadi. Un altro importante valore è il 'Search Interval', è definito come intervallo di distanza durante il processo di find dei componenti, più è elevato più il processo è rapido, più è piccolo più il processo è preciso nel definire la posizione nello spazio degli oggetti cercati. Il tuning consigliato da Fanuc è quello che rende il valore 'Search Angle Interval' tra i 5 e i 15°, nell'esempio è 4mm, corrispondenti a 10° di quest'ultimo intervallo. Come per il GPM Locator, anche questo tool prevede la possibilità di definire una finestra di ricerca in 'Search Window' (Fig 3.13b) e la possibilità di cercare il modello inclinato diversamente rispetto a quello importato. Questo è fondamentale per la ricerca di oggetti disposti in modo casuale come nel bin picking, nel tool utilizzato per i test il range in X ed Y è stato esteso al massimo, ossia 120° mentre il range in Z è sufficiente sia 60° essendo la superficie del dado esagonale. Il risultato del processo di visione con 3D One Sight Locator Tool con i componenti disposti secondo la prima configurazione (Fig 3.4a) è illustrato in Fig 3.15. Si può notare un tempo per l'esecuzione dell'intero processo di visione pari a circa 5s, in tutti i test con questi parametri e in questa configurazione i tempi rilevati sono stati sempre tra i 5 ed i 6s, oltre ad un imperfetto riconoscimento dei componenti. Questo è rappresentato dai punteggi, mai superiori a 75/100 in tutti i test eseguiti. Da questo iniziale test si potrebbe evincere che il tool in analisi ha performance peggiori del precedentemente trattato GPM, ma è stato eseguito utilizzando, come detto, le impostazioni di default. Mediante successivi test sulla stessa configurazione, è stato possibile eseguire un fine tuning del 3D Locator Tool, nello specifico modificando solo il Feature Interval portandolo a 3.3mm (91 punti), il riconoscimento dei singoli dadi è risultato più

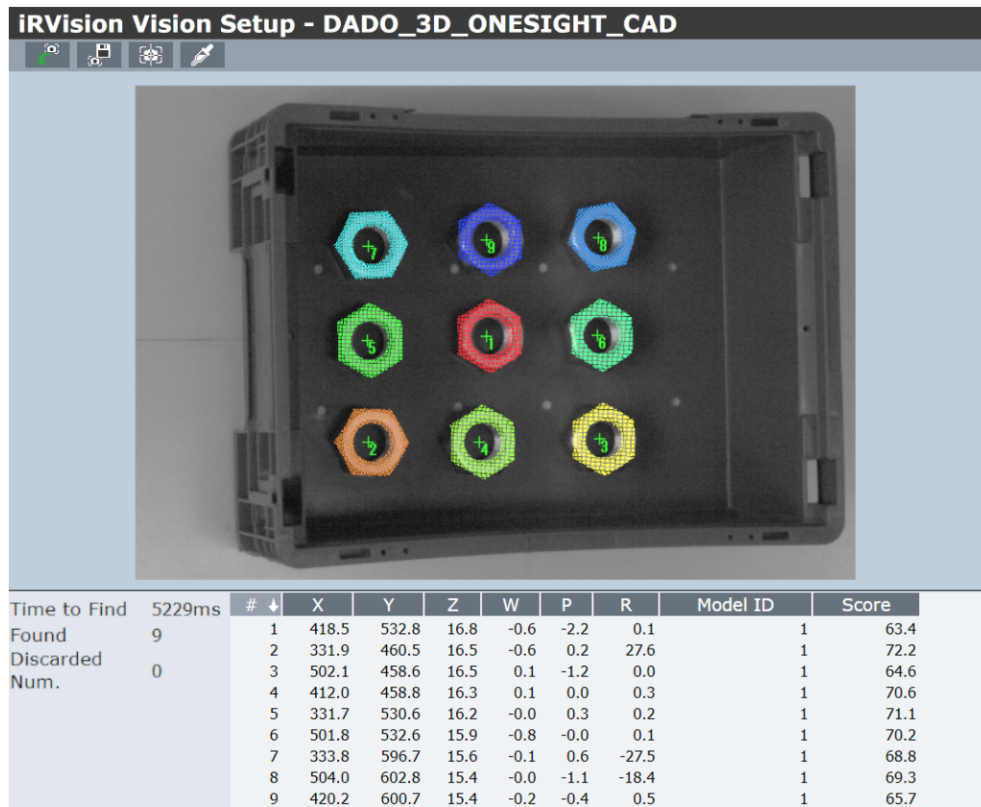


Figura 3.15: Risultato del Vision Process con 3D One Sight Model Locator Tool con i dadi disposti secondo la prima configurazione.

aderente alla realtà, lasciando invariato il tempo necessario al rilevamento, come si può vedere nel risultato sulla seconda configurazione di Fig 3.16. Si noti come i punteggi dei componenti siano mediamente più alti rispetto al test eseguito sulla precedente configurazione, inoltre il modello della vista del dado aderisce quasi perfettamente alla configurazione dei dadi riconosciuti, nonostante la loro disposizione sia più complicata. La modifica del solo parametro Feature Interval si è quindi rivelata fondamentale, in quanto permette di modificare la qualità della rilevazione dei componenti. I tempi ottenuti in tutti i test sia su prima che su seconda configurazione sono stati sempre tra i 5 ed i 6s, il Feature Interval quindi non modifica tale metrica, il parametro che più influisce su questa è il Search Interval. Riducendo anche di poco tale valore, passando quindi ad esempio da 4 a 2mm, il tempo necessario per il riconoscimento degli oggetti aumentava tanto da superare la soglia di timeout del processo di visione, impostata a 10s. Impostandolo a 3mm invece si ottenevano tempi nell'intorno degli 8s, tuttavia con risultati del tutto comparabili al settaggio di 4mm, che è quindi risultato il valore ottimo per l'applicazione analizzata. Portandolo a valori maggiori, i componenti non venivano riconosciuti correttamente, e più questo valore veniva aumentato,

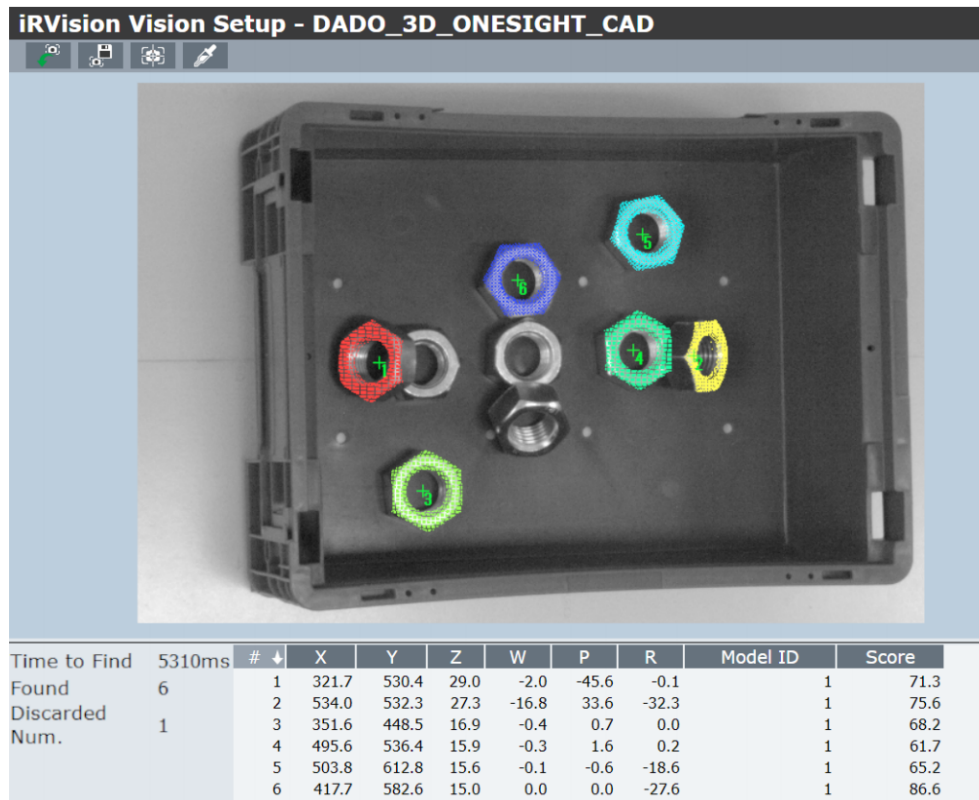


Figura 3.16: Risultato del Vision Process con 3D One Sight Model Locator Tool, modificato nel Feature Interval, con i dadi disposti secondo la seconda configurazione.

meno il modello del dado veniva riconosciuto correttamente. In Fig 3.17a,b sono illustrati i risultati del Vision Process sulla configurazione 3 e 4 rispettivamente. Nella configurazione 3 il 3D One Sight riconosce correttamente tutti e quattro i dadi, a differenza del GPM che riusciva a riconoscerne solamente uno. Nella configurazione 4 il 3D One Sight riconosce un dado, che è il più alto, quindi il risultato è corretto, a differenza del precedente tool, che come visto ne riconosceva un altro. Il problema della riflettività è sempre presente, come detto risulta indipendente dal tipo di Vision Tool utilizzato e dai relativi parametri, in quanto criticità della tecnologia del sistema di visione testato. In aggiunta, come in precedenza, il Vision Process è stato testato più volte sulle configurazioni illustrate, e a differenza del precedente GPM Locator Tool, i risultati sulla singola configurazione sono rimasti coerenti, anche al variare delle condizioni di illuminazione. Il motivo risiede nella natura stessa del tool: a differenza del GPM, Vision Tool che per il primo riconoscimento utilizza un modello 2D, il 3D One Sight è un tool che esegue un vero e proprio 3D matching, utilizzando quindi da subito un modello 3D da cercare in una mappa di punti 3D. Non si

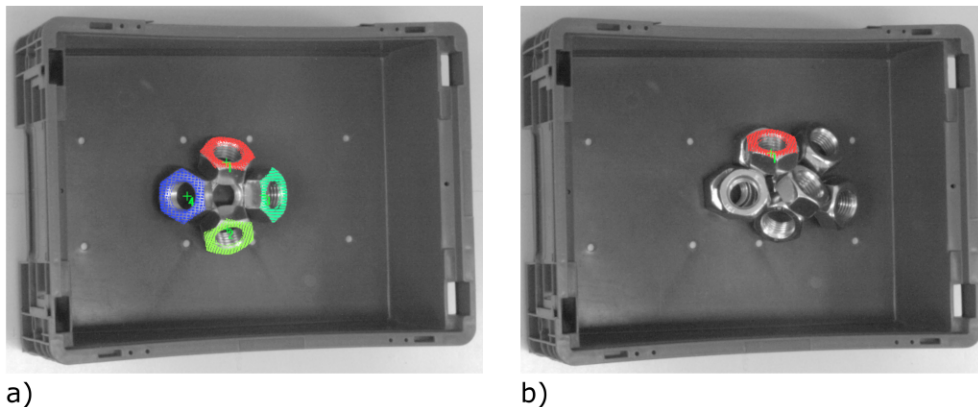


Figura 3.17: a) risultato Vision Process con 3D One Sight Locator Tool su configurazione 3, b) risultato del medesimo Vision Process su configurazione 4.

basa, infatti, su nessun tipo di immagine 2D, la quale necessita di una taratura dell'esposizione al fine di poter visualizzare correttamente gli oggetti nella scena, ma utilizza una mappa 3D, molto più robusta rispetto alle condizioni di illuminazione dell'ambiente. Questo è un risultato di fondamentale importanza, perché rende immune il processo di visione alla quantità ed al tipo di luce, che come detto sono soggette a variazioni anche significative nell'applicazione industriale.

Dopo aver testato il 3D One Sight Locator Tool con vista del componente importata da file, al fine di valutare tutte le funzionalità offerte dal medesimo tool, è stato testato il metodo di definizione del modello mediante selezione da mappa 3D. Questo è attivabile selezionando la voce 'Area' nella sezione di Model Train nell'interfaccia del medesimo Vision Tool (Fig. 3.13a). Al fine di semplificare il processo, è consigliato posizionare un singolo componente al centro del contenitore ed eseguire un'acquisizione. Dopo aver selezionato 'Area', la procedura prevede la selezione dell'area dove è presente il componente nella mappa 3D, utilizzando una finestra rettangolare modificabile dall'utente in dimensioni e posizione. Si ottiene così un insieme di punti della mappa 3D, che appartengono al componente ma anche al contenitore, per questo utilizzando successive maschere personalizzate l'utente deve fare in modo di eliminare tutti i punti che non appartengono al componente stesso. Di tale insieme di punti è possibile definire più aree, come discusso per il metodo di importazione del modello precedente, e come in quel caso, per il dado è stata sufficiente una singola zona. Il risultato ottenuto per questa applicazione è illustrata in Fig. 3.18a, dove è possibile vedere che i punti appartengono solamente alla superficie del dado. Successivamente la procedura prevede di selezionare il Feature Interval del modello appena selezionato, analogamente a quanto fatto con il metodo di selezione del modello Import File. Come in quel caso, inizialmente questo valore non è stato modificato rispetto a quanto impostato di default dal controllore. Il valore di default

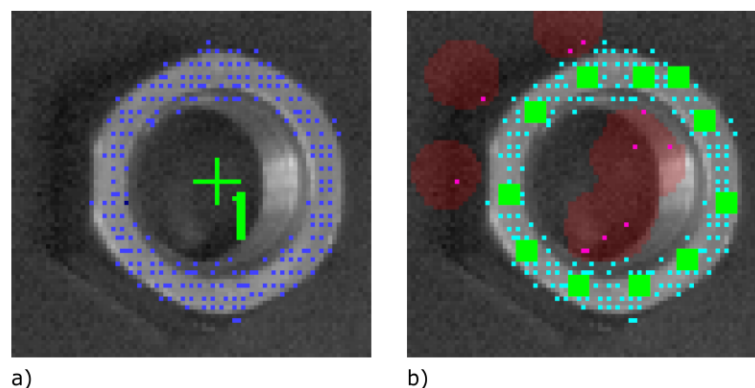


Figura 3.18: a) selezione dei punti per il model train da mappa 3D, b) in verde i punti per il Feature Interval, in rosso le maschere atte ad eliminare i punti non appartenenti al componente.

è pari a 11.8mm ed il risultato è illustrato in Fig 3.18b, dove in rosso sono evidenziate anche le maschere applicate per eliminare i punti non appartenenti al modello. Allo stesso modo, anche il Search Interval non è stato modificato rispetto al valore di default di 5mm, che si traduce in un Search Angle Interval di 11° , perfettamente all'interno del range consigliato già discusso. Come per i precedenti test, il Vision Process completo è stato testato più volte sulle già illustrate configurazioni, i risultati sulle configurazioni 1 e 2 sono riportati in Fig 3.19a,b rispettivamente. In entrambe le configurazioni ed in tutti i test eseguiti il riconoscimento del modello è avvenuto quasi perfettamente, con punteggi elevati (sempre superiori a 75/100) e tali risultati sono stati ottenuti con tempi estremamente ridotti, tra 1000 e 1300ms. Sulle configurazioni 3 e 4 i risultati sono stati analoghi a quelli illustrati in Fig 3.17a,b, ma con i tempi appena citati. L'utilizzo del modello ottenuto direttamente da una porzione di mappa 3D risulta quindi avere una precisione analoga al modello ottenuto mediante importazione di file, compresa anche la robustezza rispetto alle condizioni di illuminazione, caratteristica del 3D One Sight essendo un tool che utilizza la mappa 3D, ma impiega tempi estremamente ridotti, pari a circa 1/4 dei precedenti. Questo vantaggio nei tempi è imputabile ad un minor onere computazionale per il controllore: nel caso di modello importato da file, la vista del dado utilizzata prevede un elevato numero di punti (Fig 3.14a), 1257 per la precisione, quando nella realtà lo stesso componente visto dal sistema di visione ha 223 punti, che è ovviamente lo stesso numero che viene utilizzato nel modello da mappa 3D visto che da proviene proprio da quest'ultima. La procedura di 3D matching risulta quindi estremamente più onerosa nel caso del modello importato, portando, come visto, a tempi molto più elevati. Il parametro Feature Interval si basa su questo numero di punti, quindi anche modificandolo per rendere il numero di punti indicato tra parentesi alla destra dello stesso parametro (i punti caratte-

ristici, Fig 3.13a) non porta allo stesso onere computazionale quindi agli stessi tempi, perché appunto il numero di punti di partenza è estremamente diverso. Il Feature Interval può essere, di conseguenza, inteso come un parametro appartenente allo specifico modello selezionato quindi al suo specifico numero di punti, andando ad aumentarlo per ridurre il numero di punti caratteristici porta ad un riconoscimento degli oggetti via via meno accurato quindi ad un deterioramento dei risultati finali. Come precedentemente discusso, il numero di punti che il sistema di visione è in grado di rilevare sulla superficie del dado è inferiore a 300, estremamente limitato, questo è dovuto al fatto che questo sistema, come anticipato, è stato progettato per depallettizzazione e bin picking di oggetti di grandi dimensioni. Il dado analizzato è quindi un componente per il quale il sistema di visione non è stato progettato, viste le dimensioni ridotte e vista la riflettività della sua superficie, ed è per questo motivo che è risultato un ottimo oggetto per testare approfonditamente il sistema e verificarne i limiti.

Al fine di rimuovere la problematica della dimensione, quindi del ridotto numero di punti per componente, sono state eseguite alcune acquisizioni con delle lattine all'interno del contenitore, che hanno dimensioni molto maggiori rispetto ai dadi, ma presentano la stessa criticità del materiale riflettente. Il risultato di una di queste è già stata illustrata in Fig 3.12a,b e come si nota il numero di punti per oggetto risulta molto maggiore, tuttavia la riflettività rende comunque delle zone prive di punti.

Al fine di rimuovere invece la riflettività, sono state eseguite delle acquisizioni con degli evidenziatori all'interno del contenitore, e sono stati creati due Vision Process, aventi entrambi il 3D One Sight Locator Tool ma uno con modello da importazione ed uno con modello da area della mappa 3D, come per il dado. In Fig 3.20a,b sono riportati i due step di definizione della vista del modello e defini-

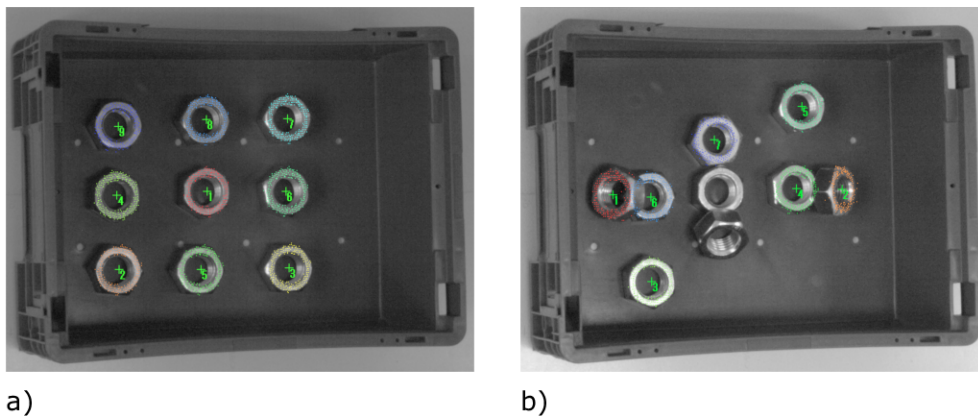


Figura 3.19: a) risultato Vision Process con 3D One Sight Locator Tool (con model teaching da area della mappa 3D) su configurazione 1, b) risultato del medesimo Vision Process su configurazione 2.

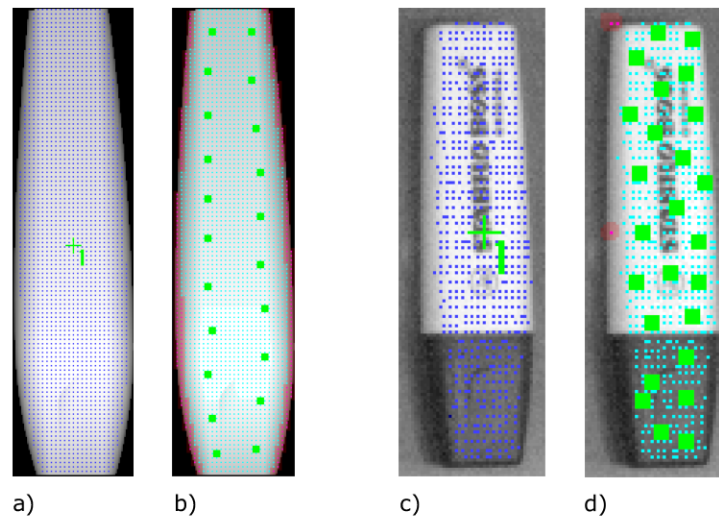


Figura 3.20: a) definizione della vista del modello evidenziatore da importazione file, b) definizione del Feature Interval sulla stessa vista, c) definizione della vista del modello da area della mappa 3D, d) definizione del Feature Interval sulla stessa vista.

zione del Feature Interval utilizzando l'importazione da file .STL. In Fig 3.20c,d sono riportati i due step di definizione della vista del modello e definizione del Feature Interval utilizzando la selezione da mappa 3D. L'evidenziatore importato da file viene considerato come unico insieme formato da 2321 punti, un numero estremamente elevato rispetto al modello ottenuto per selezione della mappa 3D, che viene considerato sempre come unico insieme ma composto da 607 punti. Tale differenza è chiaramente visibile andando a confrontare Fig 3.20a con Fig 3.20c, la densità di punti nella seconda è nettamente inferiore alla stessa nella prima. Come per i dadi, sono stati testati i Vision Process su diverse configurazioni, partendo dalle più semplici, prive di oggetti inclinati o sovrapposti, per poi aumentarne la complessità. Le impostazioni dei Vision Tools appena citati sono state modificate leggermente, a valle di numerosi test, in modo da ottimizzare e migliorare i risultati in termini di riconoscimento. Le impostazioni ottimali così definite portano il numero di punti caratteristici del modello da file a 12, mentre i punti caratteristici nel modello ottenuto da mappa 3D sono 32. In Fig 3.21a è riportato il risultato del processo di visione con modello importato da file sulla configurazione più semplice, in Fig 3.21b il risultato del processo di visione con modello ottenuto da mappa 3D, sulla medesima configurazione. In entrambi i casi sono stati riconosciuti tutti gli evidenziatori, tuttavia come era possibile prevedere dati i numeri di punti nei due modelli, il tempo impiegato dal Vision Process con importazione da file è notevolmente maggiore. Il Vision Process con modello impostato da mappa 3D ha impiegato mediamente 4500ms in tutti i test su questa configurazione, invece quello con modello da

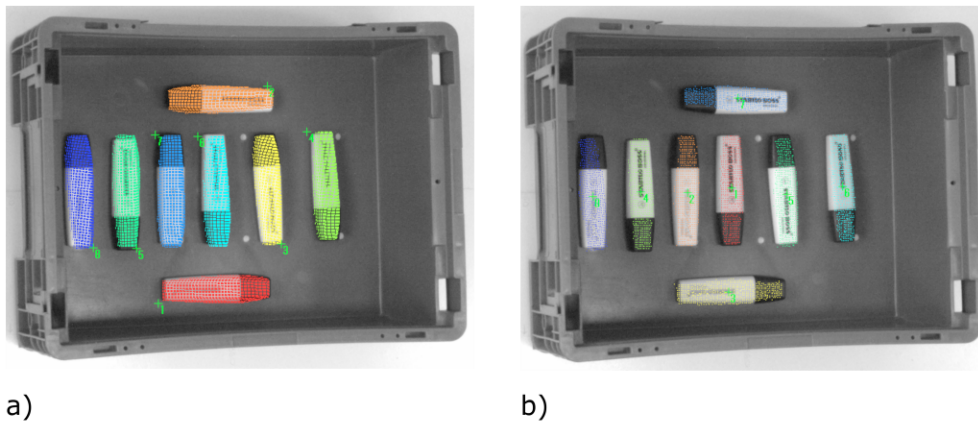


Figura 3.21: a) risultato del Vision Process con modello importato da file su configurazione 1 degli evidenziatori, b) risultato del Vision Process con modello ottenuto da mappa 3D su medesima configurazione.

file ha impiegato un tempo medio di 8100ms. È importante sottolineare che i due risultati sono del tutto confrontabili sotto il punto di vista della qualità del riconoscimento degli oggetti, la differenza sostanziale è nei tempi per ottenere tali risultati, un processo impiega quasi il doppio dell'altro. In Fig. 3.22a,b sono riportati i risultati dei due Vision Process su una delle configurazioni più complesse utilizzate per i test. Nel caso di modello importato da file (Fig. 3.22a) il processo riconosce i dadi più in alto tuttavia tale riconoscimento non è perfetto, nel caso del componente numero 2 il modello è ruotato di 180° in Z rispetto al reale evidenziatore, nel caso dei componenti 4 e 5 invece il componente non è riconosciuto con precisione. Tuttavia, i componenti inclinati vengono riconosciuti

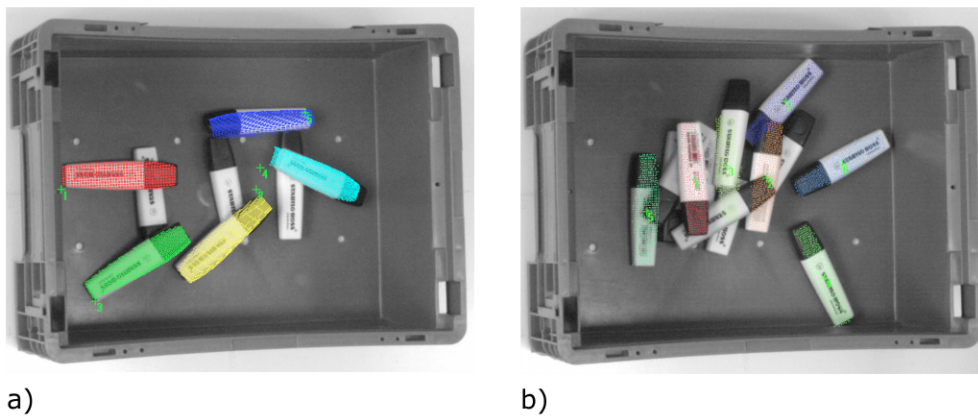


Figura 3.22: a) risultato del Vision Process con modello importato da file su configurazione 2 degli evidenziatori, b) risultato del Vision Process con modello ottenuto da mappa 3D su medesima configurazione.

tutti, a differenza dei test eseguiti con i dadi, oggetti che non riflettono, anche se di piccole dimensioni, riescono comunque ad essere visti dal sistema e ad essere riconosciuti, anche se non perfettamente. Utilizzando invece il modello ottenuto da mappa 3D, i risultati (Fig. 3.22b) mostrano un riconoscimento anche in questo caso non perfetto, ma leggermente migliore rispetto al caso precedente. I componenti vengono tutti riconosciuti correttamente ad eccezione del numero 3, che tuttavia non risulta essere un problema, in quanto durante l'applicazione reale il robot andrebbe a prelevare il componente numero 1, rendendo più semplice il rilevamento del componente 3 in un'acquisizione successiva. Come nella configurazione più semplice e in tutte le configurazioni testate, la differenza sostanziale è nei tempi per l'esecuzione del Vision Process, che nel caso di modello importato da file è mediamente 9200ms mentre nel caso del modello da mappa 3D è mediamente 6100ms. In conclusione, il sistema di visione in esame, pur non essendo adatto ad oggetti di piccole dimensioni, con un attento settaggio dei parametri dello specifico Vision Process, permette il riconoscimento anche di tali oggetti. La criticità della riflettività del materiale dei componenti o delle superfici presenti nella scena rimane sempre presente, perché, come ampiamente discusso, intrinseca del tipo di tecnologia a luce strutturata. Viene comunque citato nella manualistica Fanuc di evitare accuratamente oggetti riflettenti e trasparenti, quindi non è possibile lavorare con materiali aventi queste caratteristiche. Dai risultati ottenuti nei test, il Vision Tool più adatto per il riconoscimento 3D di oggetti messo a disposizione da Fanuc è il 3D One Sight Locator Tool, in quanto estremamente più robusto alle condizioni di illuminazione ambientale, a differenza del GPM Locator Tool, che basandosi prima sull'immagine 2D e solo poi utilizzando la mappa 3D ha performance notevolmente peggiori e pericolosamente variabili. In aggiunta, per il riconoscimento di oggetti di piccole dimensioni, il miglior metodo per la definizione della vista del componente nel 3D One Sight Locator Tool è la selezione da mappa 3D. In questo modo si ottiene un modello che ha esattamente il numero di punti che l'oggetto reale avrà nelle successive acquisizioni, il sistema si trova quindi ad eseguire il 3D matching con un modello che ha le stesse caratteristiche dell'oggetto quando questo viene acquisito nella realtà. Pur essendo un modello composto da meno punti, quindi effettivamente meno risoluto, i risultati in termini di accuratezza del riconoscimento sono perfettamente comparabili a quelli ottenuti con un modello importato, avente anche 5 volte la risoluzione, questo per il motivo appena citato. Il vantaggio principale è il tempo che impiega il Vision Process per ottenere i risultati, come visto nei test precedentemente descritti, questo risulta almeno dimezzato. Questo rappresenta una metrica importante per l'applicazione industriale, dove l'obiettivo è ridurre il più possibile il tempo ciclo, per questo nella maggior parte delle applicazioni, la computazione quindi il processo di SEARCH viene fatto eseguire mentre il robot è in movimento, in modo da evitare i tempi morti. Tuttavia è consigliabile che questo tempo rimanga sempre sotto il limite dei 10s.

Capitolo 4

Kitting motore endotermico

4.1 Introduzione

Nel seguente capitolo viene trattata un'applicazione nella quale viene impiegato il sistema descritto nel Cap.2. Dopo aver ampiamente discusso il robot comprensivo di sistema di visione, il loro funzionamento, la logica per la programmazione dei Vision Process e del controllore dell'intero sistema, averne valutato le prestazioni ed i limiti, risulta di particolare interesse applicare il tutto ad un reale processo industriale. Questo, visto la natura del documento, deve rappresentare una sfida per il sistema nella sua interezza, deve quindi avere un livello di complessità elevato per il sistema di visione con relativi Vision Process e per il controllore in termini di Interference Avoidance, Parts List Manager e programmazione del robot. L'applicazione deve andare oltre il classico bin picking in cui un robot preleva dei componenti dello stesso tipo da un singolo contenitore e li movimentata per poggiarli su un nastro trasportatore. Per questo motivo l'applicazione industriale scelta è il kitting, nello specifico la creazione di un kit per un motore endotermico. Con il termine kitting ci si riferisce all'azione di raggruppare tutti i materiali e componenti necessari per la realizzazione di un prodotto, creando così un pacchetto (chiamato appunto *kit* o *set*) che verrà consegnato agli operatori delle linee di assemblaggio che si incaricheranno dell'assemblaggio del prodotto finale o di una sua parte, allo scopo di velocizzare questa operazione. Tale processo viene eseguito in anticipo per garantire la fornitura ininterrotta di prodotti alle linee di assemblaggio. Se applicata in modo efficiente, questa strategia è in grado di apportare valore, i vantaggi derivanti dall'implementazione del kitting sono:

- Ottimizzazione dei tempi di ciclo: gli operatori delle linee di assemblaggio non devono cercare i materiali di cui hanno bisogno, poiché li hanno già tutti a portata di mano;
- Catena di approvvigionamento più dinamica: le linee di assemblaggio e il magazzino sono perfettamente collegate. Dal migliore coordinamento tra

le due aree si ottengono processi di fabbricazione più rapidi;

- Riduzione degli errori e delle inefficienze: gli operatori che preparano i kit sono specializzati in questo compito;
- Maggiore produttività: la preparazione di kit si riflette in controllo dell'inventario più rigoroso e garantisce l'assemblaggio continuo e ininterrotto delle merci;
- Migliore sfruttamento dello spazio: gli operatori nelle linee di assemblaggio dispongono solo dei prodotti e degli strumenti essenziali per svolgere il proprio lavoro. Pertanto, viene occupato lo spazio strettamente necessario per ospitare i prodotti.

Come anticipato, il kitting scelto è per un motore endotermico, più specificamente di un motore bicilindrico per motociclo, i cui principali componenti interni sono illustrati nell'assieme di Fig. 4.1. Di questi, per il kitting, sono stati selezionati quattro componenti, nello specifico i pistoni, le bielle, e le due

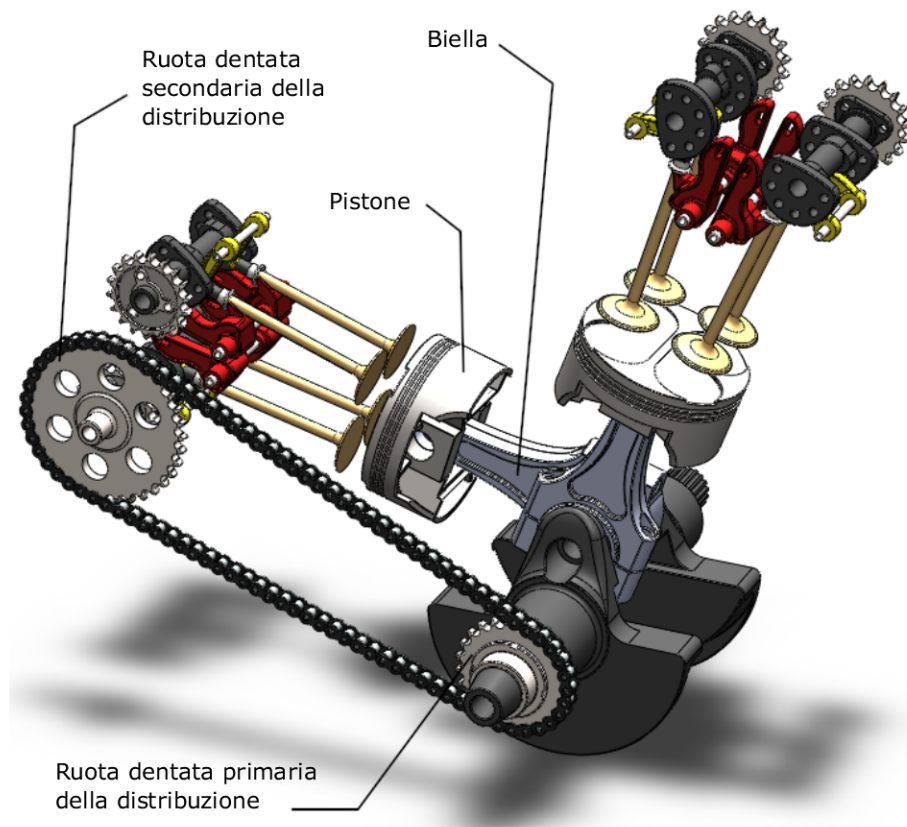


Figura 4.1: Assieme componenti interni motore bicilindrico motociclistico.

ruote dentate accoppiate dalla catena, indicati nella medesima figura. Tutti i componenti utilizzati nell'applicazione sono stati ottenuti mediante stampa 3D, utilizzando filamento in ABS di colore grigio. Inoltre i modelli sono in scala 1:2 quindi hanno dimensioni dimezzate rispetto a quelle dei componenti del motore, questo per poter utilizzare la pinza Schunk installata sul robot, che, a causa della limitata apertura delle griffe (36mm con inserti in gomma), sarebbe risultata inadatta. I quattro componenti e le loro principali dimensioni sono riportate in Fig 4.2a,b e Fig 4.3a,b.

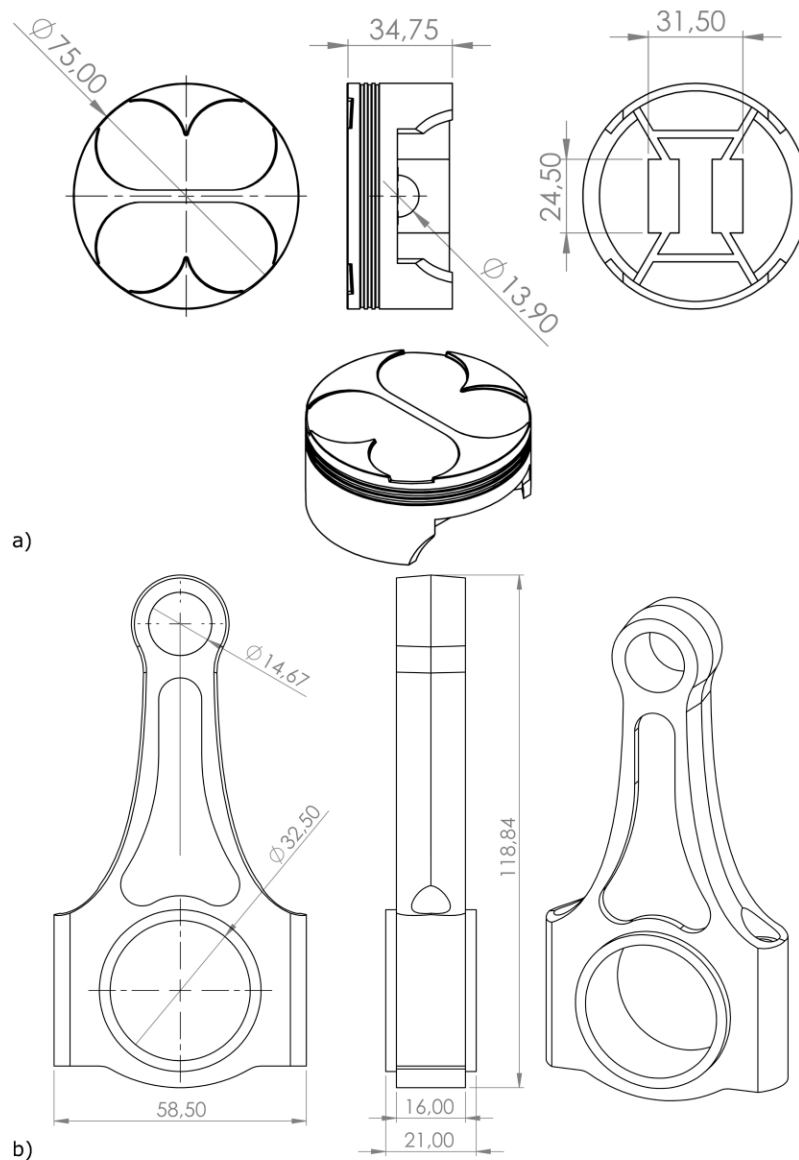


Figura 4.2: a) pistone, b) biella.

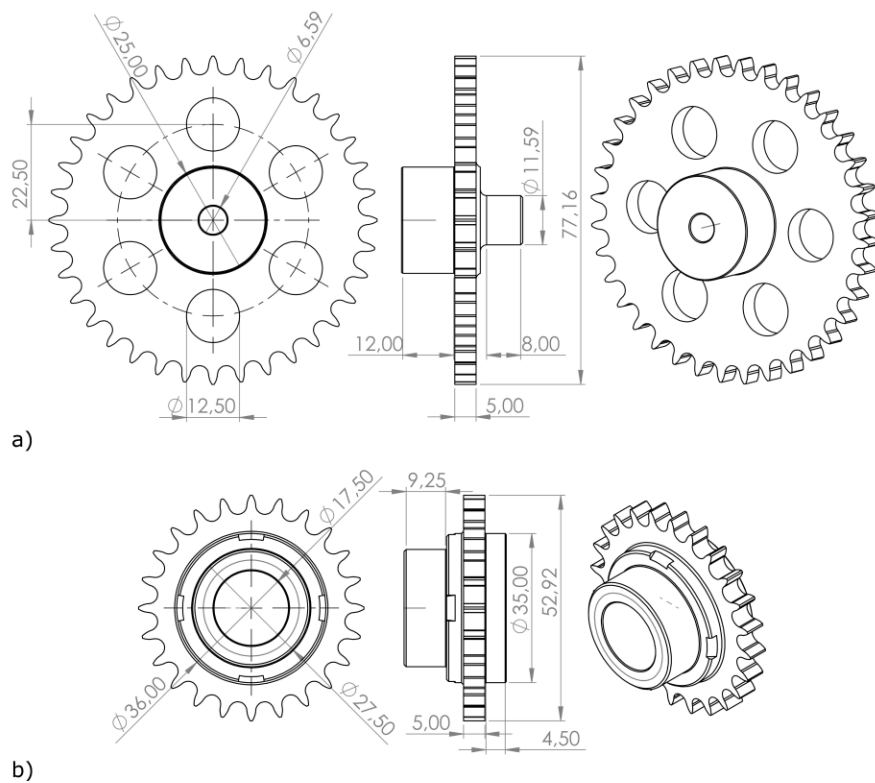


Figura 4.3: a) ruota dentata secondaria, b) ruota dentata primaria.

Sul piano di lavoro, nell'area inquadrata dal sistema di visione, sono disposti quattro contenitori, ognuno di essi contiene un tipo di componente, ad esempio vi si riferirà quindi con il nome di contenitore dei pistoni per indicare quello che contiene quello specifico tipo di pezzo, al loro interno vi è una molteplicità di pezzi dello stesso tipo. Il kit invece è composto da due componenti per ogni tipo, per un totale di otto pezzi, il robot dovrà quindi prelevare uno alla volta i componenti dagli appena citati contenitori ed eseguirne il place. Per aggiungere un livello di complessità, i componenti vengono piazzati in una specificatamente disegnata *shadow mask*, ossia un pannello di schiuma espansa avente dimensioni $350 \times 580 \times 40$ mm sul quale sono presenti le sagome dei componenti facenti parte del kit. Il robot deve quindi eseguire un place preciso di ogni componente, visto che ognuno di essi ha il suo specifico alloggiamento all'interno della *shadow mask*. Al termine del ciclo, il robot ha riempito tutti e otto gli alloggiamenti della *shadow mask* con gli otto componenti che fanno parte del kit per il motore endotermico. Ad ogni pezzo preso dal robot è necessario eseguire una nuova scansione con relativo processo di SEARCH perché i componenti nell'intorno di quello preso sono sicuramente stati spostati quindi la loro posizione ed orientazione non è più quella rilevata in precedenza. Al fine di ottimizzare i tempi, quindi ridurre il tempo ciclo, la scansione+SEARCH che seguono la presa di un

componente vengono eseguiti mentre il robot effettua il place, per la precisione non appena questo esce dal campo inquadrato dalle telecamere. In questo modo si evita che il robot rimanga in attesa della scansione e del risultato della ricerca del successivo componente da prelevare. L'applicazione di kitting così composta risulta essere particolarmente impegnativa per il sistema Fanuc nella sua interezza in quanto è necessario gestire quattro tipi diversi di componenti, quattro distinti contenitori, ed il kit nel suo complesso, ossia come numero, tipo di componenti e loro disposizione nel kit stesso. La cella atta ad eseguire il kitting è illustrata in Fig 4.4

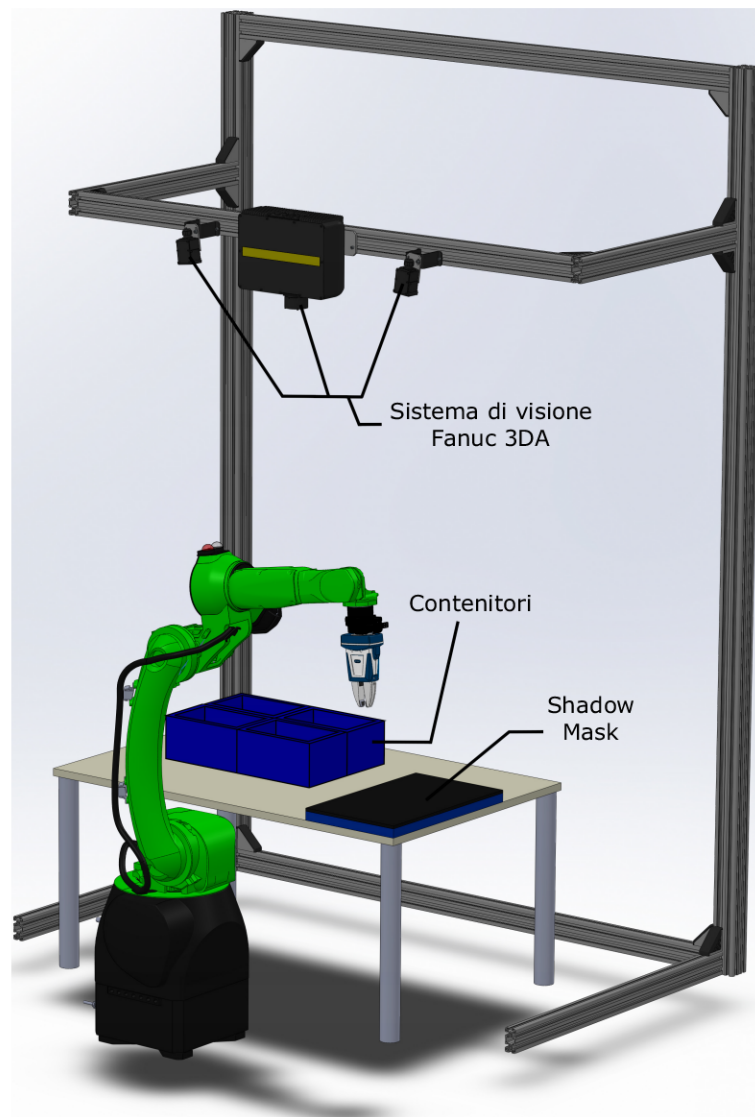


Figura 4.4: Cella robotizzata per il kitting analizzato.

4.2 Vision Process setting

Per il riconoscimento dei quattro componenti precedentemente descritti sono necessari quattro distinti processi di visione, uno per componente. Ogni Vision Process è di tipo 3D Area Sensor con una struttura come quella illustrata in Fig. 4.5, ciò che cambia è nel 3D One Sight Model Locator Tool, nel quale si ha lo specifico modello da riconoscere. Nella stessa figura è riportato il Vision Process per il riconoscimento dei pistoni, dove nel 3D One Sight si ha il modello del medesimo componente. Lo Snap Tool presente in tutti i processi di visione

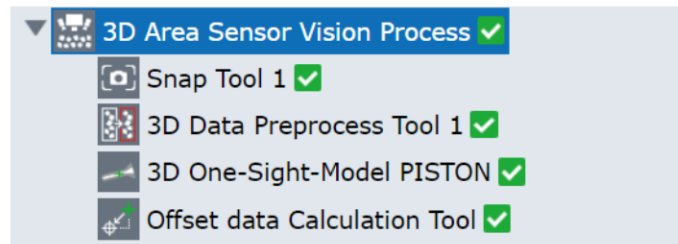


Figura 4.5: Struttura del Vision Process atto al riconoscimento dei pistoni.

è impostato in modo che l'immagine scattata dal sistema di visione sia solo del contenitore d'interesse per quello specifico Vision Process. Lo Snap Tool presente in Fig. 4.5 è relativo al processo per il riconoscimento dei pistoni quindi l'area selezionata all'interno dell'intera immagine è limitata al contenitore del medesimo componente, in questo modo si riduce il tempo di calcolo per il controllore. In Fig. 4.6a è riportato l'intero FOV del sistema di visione, in Fig. 4.6b la porzione utile al Vision Process per il riconoscimento delle bielle, ottenuta grazie al settaggio della finestra dello Snap Tool. Nel 3D Data Preprocess Tool è necessario impostare il contenitore, indicando i suoi quattro vertici e la sua

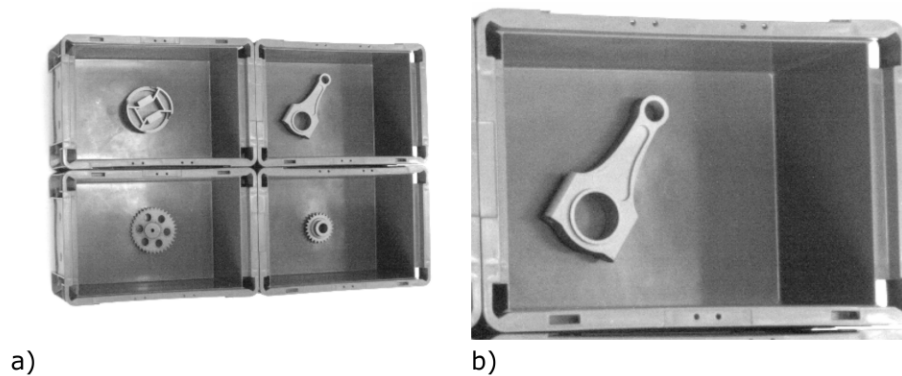


Figura 4.6: a) immagine completa catturata dal sistema di visione, b) immagine utilizzata nel processo di visione per il riconoscimento delle bielle, ottenuta grazie alla riduzione dell'immagine attuata dallo Snap Tool.

profondità. Così facendo i successivi tool che utilizzano la mappa 3D non cercano l'oggetto in tutta la mappa ma solo all'interno dello specificato contenitore, che come per lo Snap Tool, è differente per ogni componente quindi per ogni processo di visione. Anche questo settaggio permette di ridurre notevolmente i tempi di calcolo.

Il tool più importante nei processi di visione analizzati è sicuramente il 3D One Sight Model Locator Tool, il quale differisce nei Vision Process per il tipo di oggetto da riconoscere e per i settaggi dei parametri per il modello specifico. Sono stati eseguiti numerosi test su tutte e quattro le tipologie di componenti, al fine di valutare l'opzione migliore tra le già discusse importazione del modello da file e la selezione da mappa 3D dello stesso, nel 3D One Sight Locator. Nei test descritti nel Cap.3.2.2 l'opzione migliore tra le due è emersa essere la selezione da mappa 3D, in quanto permette di ottenere ottimi risultati con tempi nettamente inferiori rispetto alla selezione della vista del modello importando il file dello stesso. Tuttavia per i componenti utilizzati nel kit, la selezione da mappa 3D è risultata fallimentare, nello specifico le viste selezionate per il riconoscimento dei componenti sono state valutate troppo complesse per il tool, portando all'impossibilità di utilizzo del tool stesso. Optando invece per la selezione della vista del modello da importazione di file, il 3D One Sight Locator Tool funziona correttamente e, come sarà discusso in seguito, porta a degli ottimi risultati sotto il punto di vista del riconoscimento. In ogni 3D One Sight Locator Tool è stato quindi importato il modello relativo al componente da riconoscere, che coincide con il modello utilizzato per la stampa 3D del componente stesso. Di ogni componente è stata selezionata una vista, come previsto dall'appena citato tool, questa deve avere una caratteristica che renda possibile la presa del pezzo con la pinza in dotazione nella cella robotizzata. In Fig.4.7 sono riportate le viste dei componenti che i processi di visione utilizzano per eseguire il 3D matching, in particolare in Fig.4.7a,c,e,g sono illustrati i punti considerati per ogni vista dei modelli mentre in Fig.4.7b,d,f,h si ha la rappresentazione del Feature Interval ed in rosso è evidenziata la maschera applicata per rimuovere i punti non significativi al fine del riconoscimento. Dal modello in formato .STL importato è stata selezionata, come detto, una vista, e su questa è stato necessario apportare modifiche, atte al miglioramento del riconoscimento e alla diminuzione del tempo computazionale ad esso relativo. Ad ogni componente è associato un ID, numero identificativo univoco, utile successivamente nelle macro del programma TP. Vengono elencati gli accorgimenti applicati ad ogni vista selezionata:

- Pistone: riportato in Fig.4.7a,b, dopo aver selezionato la vista illustrata nella medesima figura, ed aver considerato tutti i punti riconosciuti appartenenti ad un unico insieme, il controllore applica di default una maschera che elimina i punti appartenenti alla parte più interna del modello ossia la parte sottostante al cielo del pistone stesso. Tale maschera è fondamentale in quanto del pistone reale il sistema di visione riesce ad ottenere informazioni solo riguardanti la superficie più esposta. Per questo è consigliato

quindi eliminare i punti che si prevede non avranno dei corrispondenti nella mappa 3D del componente reale. Tale maschera applicata di default è stata modificata ed allargata ulteriormente, andando ad eliminare altri punti appartenenti ad altre superfici che il sistema di visione non riesce a rilevare correttamente. Il modello definitivo utilizzato per il 3D matching è illustrato in Fig. 4.7a, la maschera applicata per la rimozione dei punti appena descritta è illustrata in rosso in Fig. 4.7b, nella stessa immagine sono rappresentati in verde i punti del Feature Interval. Il numero identificativo di questo componente è 1;

- Biella: riportata in Fig. 4.7c,d, anche questo modello è stato considerato come unico insieme di punti, la maschera applicata di default dal controllore è

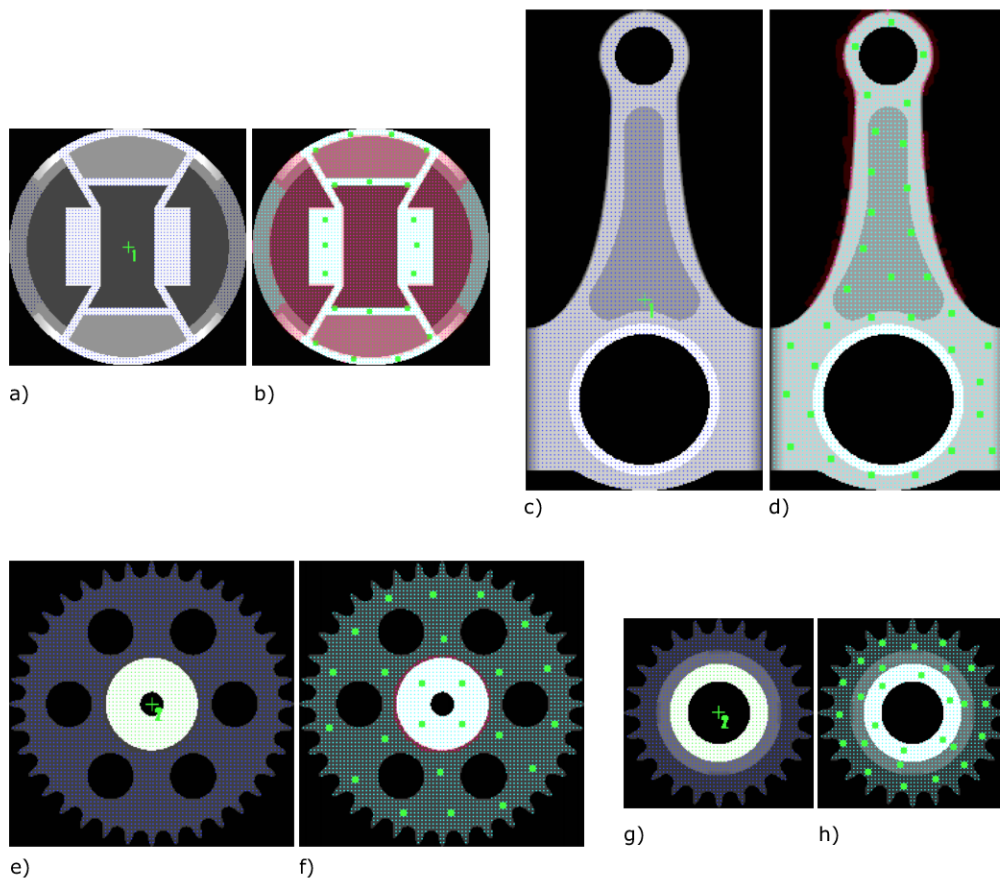


Figura 4.7: a,c,e,g) vista del modello pistone, biella, ruota dentata secondaria e primaria rispettivamente, con relativi punti (in blu) considerati dal Vision Tool per il 3D matching, b,d,f,h) vista del modello con maschera (in rosso) per eliminare i punti non rilevanti per il riconoscimento e punti appartenenti al Feature Interval (quadrati verdi).

stata eliminata in quanto portava a risultati non accurati di riconoscimento. Sono stati esclusi i punti più vicini al perimetro della vista del modello mediante una successiva maschera personalizzata. Questo perché tali punti sono stati riconosciuti come ad altezze leggermente diverse da quelli appartenenti alla superficie principale e per questo durante il 3D matching il riconoscimento non avveniva con precisione. Il numero identificativo di questo componente è 2;

- Ruota dentata secondaria: riportata in Fig. 4.7e,f, a differenza dei modelli discussi in precedenza, è stato particolarmente conveniente dividere la vista del modello in due insiemi di punti, uno rappresentante il toroide evidenziato con i punti verdi in Fig. 4.7e ed uno che include tutti gli altri punti, composto dai punti blu nella medesima figura. I centri dei due insiemi di punti coincidono nella croce di colore verde, i numeri 1 e 2 sono risultano per questo sovrapposti e rappresentano il centro dei due insiemi. La divisione è stata necessaria perché il componente ha superfici appartenenti a piani diversi, come illustrato in Fig. 4.3a, quindi di fatto sono rilevate dal sistema di visione come distinte. In aggiunta è stato definito come *Emphasis Area* il secondo insieme di punti appena descritto. Questa funzione permette di eseguire una ricerca dapprima dell'area in essa impostata, e solo poi degli altri insiemi di punti. I vantaggi che porta questa impostazione si sono rivelati significativi, ha infatti aumentato la precisione di riconoscimento, diminuendo il tempo impiegato per lo stesso. La maschera in rosso di Fig. 4.7f è stata applicata per eliminare i punti appartenenti alla giunzione tra il primo ed il secondo insieme, perché questi potrebbero risultare coperti dalla parte estrusa del componente stesso nelle acquisizioni. Il numero identificativo di questo componente è 3;
- Ruota dentata primaria: riportata in Fig. 4.7g,h, ha subito una divisione in insiemi di punti analoga alla precedente ruota dentata, viste le caratteristiche comuni a quel componente, ottenendo i medesimi vantaggi. Questo modello non necessita alcuna maschera atta alla rimozione di punti. Il numero identificativo di questo componente è 4.

Per tutti i componenti è fondamentale sottolineare che la posizione ed orientazione che si ottiene come output dai Vision Process è riferita al sistema di riferimento del modello importato. Tale posizione ed orientazione sarà la medesima che il robot assumerà per eseguire la presa del componente, perché il sistema Fanuc prevede di sovrapporre perfettamente il sistema di riferimento del Tool con il sistema di riferimento dell'oggetto. Per questo, il sistema di riferimento dell'oggetto, impostato durante la conversione da file SolidWorks a file STL, deve rendere possibile il pick dello stesso componente. A tal fine per ogni oggetto è stato inserito all'interno del corpo, ad una distanza di 5mm dalla superficie della vista selezionata, ed in una posizione nella quale fosse possibile per la pinza Schunk chiudere le griffe ed afferrare il componente. I parametri

Feature Interval e Search Interval hanno subito un fine tuning, al fine di ridurre il più possibile il Search Angle Interval, con l'obiettivo di renderlo di un valore interno al range consigliato 5° - 15° , senza portare i tempi di riconoscimento sopra i 7s. Per tutti i componenti i range angolari di ricerca del modello rispetto agli assi X, Y, Z sono stati impostati ai valori massimi consentiti ossia $\pm 60^{\circ}$, $\pm 60^{\circ}$ e $\pm 180^{\circ}$ rispettivamente, in modo da poter riconoscere anche i componenti con orientazioni più estreme.

4.3 Interference Avoidance setting

L'applicazione analizzata prevede, come anticipato, l'utilizzo di quattro differenti contenitori, come ampiamente discusso nel Cap [2.4.2](#), è necessario impostare la loro disposizione e le loro dimensioni come Interference Avoidance Setup System. Solo in questo modo il controllore potrà eseguire un corretto controllo delle collisioni, in aggiunta anche il piano di lavoro è stato incluso in tale Interference Avoidance, aggiungendo una sicurezza ulteriore. Per l'impostazione dei contenitori e del piano di lavoro, rappresentato da un tavolo $1400 \times 700 \times 700$ mm, è stato utilizzato il three way method descritto nel precedentemente citato capitolo. Ad ogni contenitore è stato associato un nome ed un ID univoco, *CONTAI-*

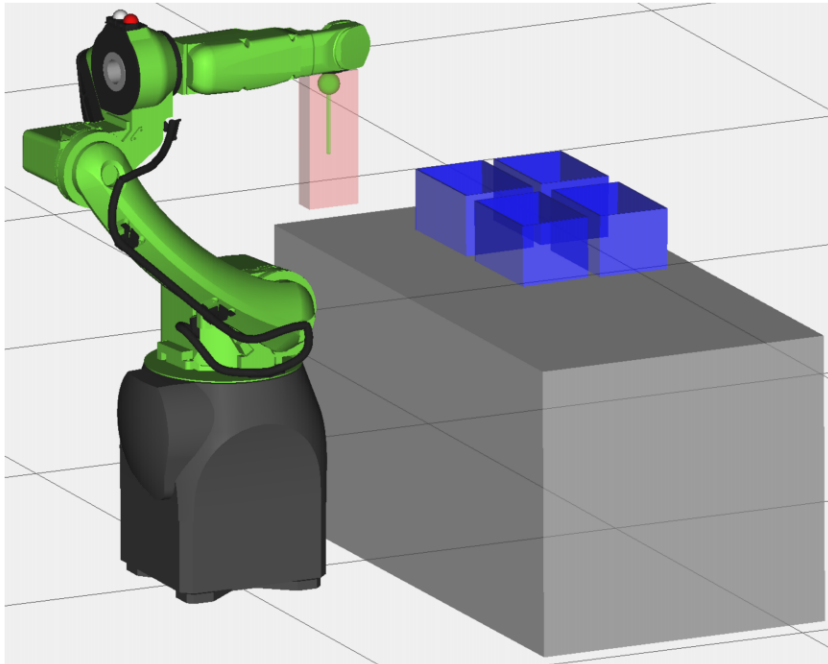


Figura 4.8: Cella robotizzata considerata per il controllo delle collisioni, in grigio il modello rappresentante piano di lavoro, in blu i modelli per i quattro contenitori ed in rosso il modello per il gripper.

NER_PISTONS ed 1 per il contenitore dei pistoni, *CONTAINER_CONRODS* e 2 per quello delle bielle, *CONTAINER_SPROCKBIG* e 3 per quello delle ruote dentate secondarie, *CONTAINER_SPROCKSM* e 4 per quello delle ruote dentate primarie. In questo modo è possibile distinguere le scatole nelle successive sezioni di controllo delle interferenze e Parts List Manager. Non è stato necessario includere il supporto del sistema di visione in quanto posizionato al di fuori dello spazio di lavoro del robot, risultando quindi non raggiungibile dal manipolatore. La pinza è stata inclusa in un Interference Avoidance Setup Robot, di tipo HAND e rappresentata da un esaedro. I modelli ottenuti dalle impostazioni dalle appena discusse impostazioni per l'Interference Avoidance sono illustrati in Fig. 4.8. Le impostazioni dell'Interference Avoidance Setup Condition sono riportate in Fig. 4.9, in cui si nota il massimo angolo consentito tra asse Z del tool e asse Z dell'oggetto pari a 65° , il metodo di calcolo e selezione della posizione calcolata per evitare la collisione considera la minima traslazione dell'end effector rispetto all'oggetto da afferrare. Il range di rotazione rispetto all'asse Z è stato impostato al massimo valore in quanto la maggior parte degli oggetti considerati hanno superfici circolari quindi il pick dell'oggetto avviene con successo anche se il tool ruota rispetto a quell'asse.

Axis	Enable	Minimum	Maximum
X	<input type="checkbox"/>	0.0	0.0 mm
Y	<input type="checkbox"/>	0.0	0.0 mm
W	<input type="checkbox"/>	-40.0	40.0 °
P	<input type="checkbox"/>	-90.0	90.0 °
R	<input checked="" type="checkbox"/>	-180.0	180.0 °

Figura 4.9: Interfaccia di impostazione per i parametri relativi all'Interference Avoidance Setup Condition utilizzata per il kitting.

4.4 Parts List Manager setting

Dovendo gestire quattro diversi componenti e quattro contenitori è stato scelto di utilizzare un'unica Parts List, nella quale sono presenti quattro processi di visione, uno per componente, e quattro PICK Position List, per la gestione delle scatole. Nello specifico, la VP List è riportata in Fig 4.10 ed è composta dai quattro processi di visione presenti nella medesima immagine. Il pro-

SEARCH VP List Parts List1			
#	Vision Process Name	Img. Reg	Priority
1	A_PISTON_3D_ONESIGHT_CAD	0	Measurement1
2	A_CONNROD_3D_ONESIGHT_CAD	0	Measurement1
3	A_SPROCKETBIG_3D_ONESIGHT_CAD	0	Measurement1
4	A_SPROCKETSMALL_3D_ONESIGHT_CAD	0	Measurement1

Figura 4.10: Interfaccia della Vision Process List nella Parts List.

cesso di visione *A_PISTON_3D_ONESIGHT_CAD* è atto al riconoscimento dei pistoni, *A_CONNROD_3D_ONESIGHT_CAD* al riconoscimento delle bielle, *A_SPROCKETBIG_3D_ONESIGHT_CAD* al riconoscimento delle ruote dentate secondarie e *A_SPROCKETSMALL_3D_ONESIGHT_CAD* al riconoscimento delle ruote dentate primarie. Ciascun processo nella lista ha un numero identificativo univoco (VP ID) che, per comodità nella successiva programmazione, è lo stesso dell'ID del componente da rilevare nello specifico Vision Process. Tali ID corrispondono anche con l'ordine dei componenti con il quale il kit viene composto. Analogamente, all'interno della medesima Parts List, la PICK Position List è composta da quattro elementi, anche in questo caso uno per componente. Tale lista, come discusso nel Cap 2.4.3, gestisce i Vision Process con l'Interference Avoidance, è anch'essa suddivisa in quattro, con quattro numeri identificativi univoci (PICK Position List ID). Una volta richiamata dal-

PICK Position List Parts List1			
#	Comment	Vision Process Name	Interference Setup
1	Pick Pistoni	A_PISTON_3D_ONESIGHT_CAD	1 (CONTAINER_PISTONS,SCHUNK_GRIPPER,INTERF_DR)
2	Pick Bielle	A_CONNROD_3D_ONESIGHT_CAD	2 (CONTAINER_CONRODS,SCHUNK_GRIPPER,INTERF_DR)
3	Pick ruota dentata secondaria	A_SPROCKETBIG_3D_ONESIGHT_CAD	3 (CONTAINER_SPROCKETBIG,SCHUNK_GRIPPER,INTERF_DR)
4	Pick ruota dentata primaria	A_SPROCKETSMALL_3D_ONESIGHT_CAD	4 (CONTAINER_SPROCKETSM,SCHUNK_GRIPPER,INTERF_DR)

Comment	Pick Pistoni
Use Found Position	Enable <input type="checkbox"/>
Vision Process Name	A_PISTON_3D_ONESIGHT_CAD Trained Riconoscimento Pistoni
Model ID	1
Interference Setup	
Calculate IA	Enable <input checked="" type="checkbox"/>
IASYS	CONTAINER_PISTONS UF:2
IAROB	SCHUNK_GRIPPER
IACND	INTERF_DR UT:4
Approach Setup	
IACND	INTERF_DR UT:4
Ofs	PR 0 Not Used
Tofs	PR 100

Figura 4.11: Interfaccia della PICK Position List nella Parts List.

la macro per il calcolo della posizione di pick, mediante lo specifico ID appena citato, questa viene calcolata facendo riferimento allo specifico Vision Process, allo specifico ID componente ed agli specifici parametri di Interference Avoidance indicati. In Fig. 4.11 è riportata in alto la PICK Position List ed in basso il dettaglio del primo elemento della medesima lista. Anche in questo caso l'ID corrisponde all'ID del componente, quindi il dettaglio è relativo all'elemento pistone. Per questo il processo di visione scelto è il *A_PISTON_3D_ONESIGHT_CAD*, l'ID componente è pari a 1, l'Interference Avoidance Setup System (IASYS) è il *CONTAINER_PISTONS*, il Setup Robot (IAROB) è *SCHUNK_GRIPPER* e la Condition (IACND) è *INTERF_DR*. Ogni elemento della PICK Position List differisce dagli altri per il Vision Process Name, Model ID (perché legati al tipo di componente da riconoscere) e IASYS (perché legato al contenitore da considerare nel calcolo delle collisioni). Alle impostazioni di Push Part Data Setup e Status Setup List non sono state apportate modifiche rispetto ai loro valori di default.

4.5 TP Programming

Nel Cap[2.4.3] è stato approfonditamente descritto il programma base per il bin picking nel sistema Fanuc, tuttavia al fine di testare approfonditamente il controllore, per il kitting finora descritto è stato utilizzato e modificato il programma di bin picking avanzato, la cui logica è illustrata in Fig[4.12]

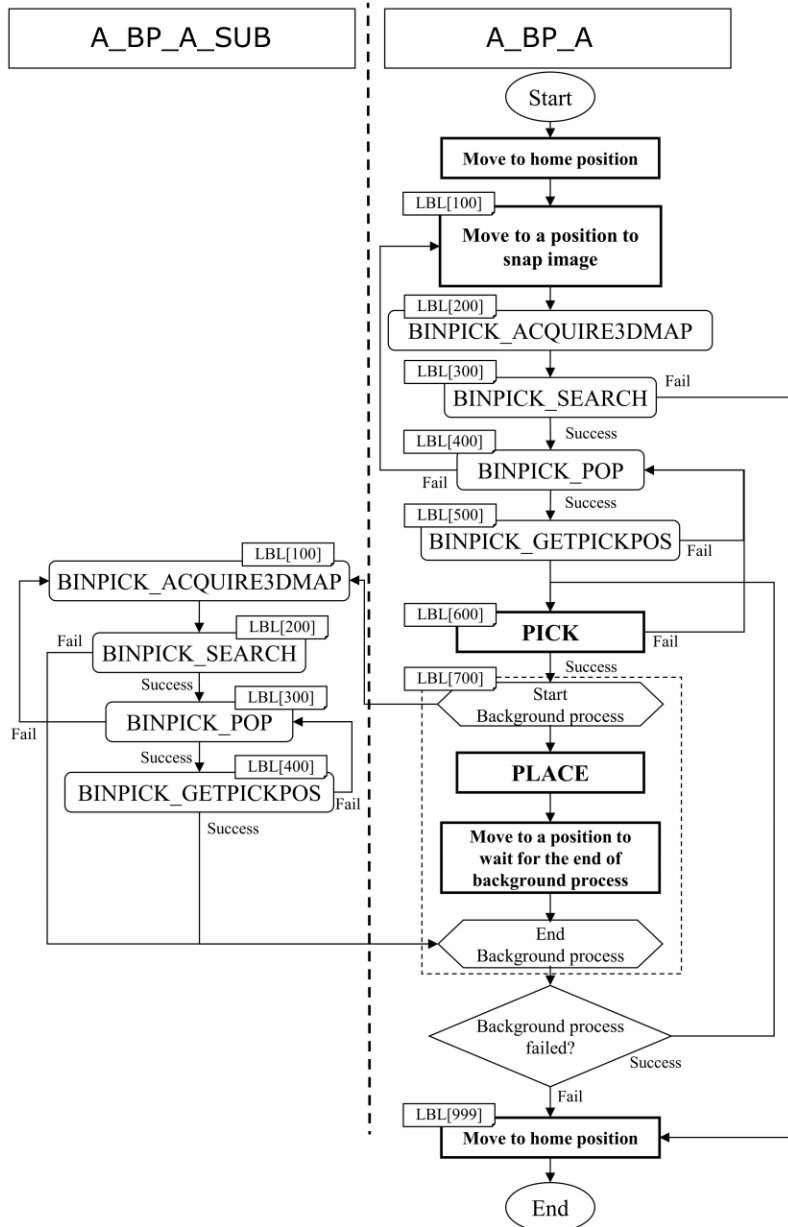


Figura 4.12: Flowchart del programma TP di bin picking utilizzato nel kitting per motore endotermico.

Il programma in questione è composto da un programma principale (main), indicato con *A_BP_A*, ed un sottoprogramma (subprogram), *A_BP_A_SUB*. Il main ha la stessa logica del programma di base descritto in precedenza, ossia porta il robot ad una posizione di Home, poi ad una posizione per eseguire un'acquisizione (può essere la stessa Home), viene eseguita un'acquisizione 3D dal sistema di visione, in questa vengono cercati i componenti (SEARCH) secondo lo specificato Vision Process, i componenti rilevati vengono aggiunti alla Parts List, da questa viene estratto il primo e ne viene calcolata posizione ed orientazione per la presa da parte del manipolatore. Successivamente quest'ultimo esegue la presa e va a poggiare il componente nella posizione di place impostata. Tuttavia, come si vede dalla Fig. 4.12, dopo aver eseguito il pick del componente designato, viene avviato in parallelo il sottoprogramma, il quale attende che il robot sia fuori dal FOV del sistema di visione, esegue una nuova acquisizione 3D, seguita da un nuovo processo di SEARCH, aggiungendo alla Parts List i componenti riconosciuti, per poi estrarne quello a priorità maggiore e calcolarne la posizione ed orientazione. In questo modo, l'acquisizione e tutta la computazione per il rilevamento dei componenti viene eseguita mentre il robot esegue il place, eliminando i tempi morti quindi il tempo per un ciclo completo. Di seguito vengono riportati il main ed il subprogram completi.

A_BP_A (main):

```

1 J P[1:Home] 100% FINE
2 CALL GRIP_OPEN
3 !
4 !Initialize Data
5 R[110]=0
6 R[111]=0
7 R[112]=0
8 R[113]=0
9 R[114]=0
10 R[115]=0
11 R[116]=0
12 R[120]=0
13
14
15 CALL BINPICK_CLEAR("Parts List ID"=1)
16
17 R[100]=1
18
19 UFRAME_NUM=2
20 UTOOL_NUM=4
21
22 !Move to Search Pos
23 LBL[100]
24 !SEARCH Pos
25 J P[2:SEARCH] 100% FINE
26 !Robot NOT in FOV
27 R[117]=1
28
29 !ACQUIRE 3D MAP
30 LBL[200]
31 CALL BINPICK_ACQUIRE3DMAP("3D Area Sensor"='FIXED')
32
33 !SEARCH
34 LBL[300]
35 CALL BINPICK_SEARCH("Parts List ID"=1,"SEARCH VP ID"=R[100],"Status R"=1
36 13)
37 IF R[113]=0,JMP LBL[400]
38 !else = SEARCH Fail
39 R[110]=1
40 JMP LBL[999]
41
42 !POP
43 LBL[400]
44 CALL BINPICK_POP("Parts List ID"=1,"Status R"=114,"Model ID R"=115)
45 IF R[114]=0,JMP LBL[500]
46 !else = POP Fail
47 JMP LBL[100]
48
49 !GETPICKPOS
50 LBL[500]
51 CALL BINPICK_GETPICKPOS("Parts List ID"=1,"PICK Pos ID"=R[100],"Status R
52 "=116,"PICK Pos PR"=50,"IA Offset PR"=51,"APP Pos PR"=52)
53 IF R[116]=0,JMP LBL[600]
54 !else = GETPICKPOS Fail
55 CALL BINPICK_SETSTAT("Parts List ID"=1,"Part Status"=22)
56 JMP LBL[400]
57
58 !PICK
59 LBL[600]
60 !ROBOT in FOV

```

```

59 R[117]=0
60 !Move above the container
61 L P[3:Above Containers] 500mm/sec CNT50
62 !Approach
63 L PR[52] 500mm/sec CNT50
64 !Pick Pos
65 L PR[50] 200mm/sec FINE
66 !Insert program instructions
67 !to grasp the part.
68 CALL GRIP_CLOSE
69
70 !components update
71 R[120]=R[120]+1
72
73 IF R[119]=1,JMP LBL[610]
74 !else=PICK Fail
75 CALL BINPICK_SETSTAT("Parts List ID"=1,"Part Status"=21)
76 JMP LBL[400]
77
78 !PICK Success
79 LBL[610]
80 CALL BINPICK_SETSTAT("Parts List ID"=1,"Part Status"=20)
81 !Depart
82 L PR[52] 500mm/sec CNT50
83 !Move above the container
84 L P[3:Above Containers] 500mm/sec CNT50
85
86 !Start Background Calc. and PLACE
87 LBL[700]
88 !start subprogram
89 R[112]=0
90
91 !if 8 wp placed, end
92 IF R[120]=8,JMP LBL[800]
93 RUN A_BP_A_SUB
94
95 !Move out from FOV
96 J P[4:Intermediate] 100% CNT100 TA .50sec,CALL A_FOV_A
97
98 !PLACE pistone 1
99 IF (R[120]=1) THEN
100 !PLACE Approach
101 L PR[80] 300mm/sec CNT100
102 !PLACE
103 L PR[81] 300mm/sec FINE
104 CALL GRIP_OPEN
105 L PR[80] 300mm/sec CNT100
106 ENDIF
107
108 !PLACE pistone 2
109 IF (R[120]=2) THEN
110 !PLACE Approach
111 L PR[82] 300mm/sec CNT100
112 !PLACE
113 L PR[83] 300mm/sec FINE
114 CALL GRIP_OPEN
115 L PR[82] 300mm/sec CNT100
116 ENDIF
117
118 !PLACE biella 1
119 IF (R[120]=3) THEN
120 !PLACE Approach

```

```
121 L PR[84] 300mm/sec CNT100
122 !PLACE
123 L PR[85] 300mm/sec FINE
124 CALL GRIP_OPEN
125 L PR[84] 300mm/sec CNT100
126 ENDIF
127
128 !PLACE biella 2
129 IF (R[120]=4) THEN
130 !PLACE Approach
131 L PR[86] 300mm/sec CNT100
132 !PLACE
133 L PR[87] 300mm/sec FINE
134 CALL GRIP_OPEN
135 L PR[86] 300mm/sec CNT100
136 ENDIF
137
138 !PLACE ingranaggio primario 1
139 IF (R[120]=5) THEN
140 !PLACE Approach
141 L PR[88] 300mm/sec CNT100
142 !PLACE
143 L PR[89] 300mm/sec FINE
144 CALL GRIP_OPEN
145 L PR[88] 300mm/sec CNT100
146 ENDIF
147
148 !PLACE ingranaggio primario 2
149 IF (R[120]=6) THEN
150 !PLACE Approach
151 L PR[90] 300mm/sec CNT100
152 !PLACE
153 L PR[91] 300mm/sec FINE
154 CALL GRIP_OPEN
155 L PR[90] 300mm/sec CNT100
156 ENDIF
157
158 !PLACE ingranaggio secondario 1
159 IF (R[120]=7) THEN
160 !PLACE Approach
161 L PR[92] 300mm/sec CNT100
162 !PLACE
163 L PR[93] 300mm/sec FINE
164 CALL GRIP_OPEN
165 L PR[92] 300mm/sec CNT100
166 ENDIF
167
168 LBL[800]
169 !PLACE ingranaggio secondario 2
170 IF (R[120]=8) THEN
171 !PLACE Approach
172 L PR[94] 300mm/sec CNT100
173 !PLACE
174 L PR[95] 300mm/sec FINE
175 CALL GRIP_OPEN
176 L PR[94] 300mm/sec CNT100
177 !program end
178 JMP LBL[999]
179 ENDIF
180
181 WAIT R[112]=1
182 !End Background Calc.
```

```

183
184 !Sys Status check
185 IF R[110]=1,JMP LBL[999]
186
187 !Program Loop
188 JMP LBL[600]
189
190 !Program END
191 LBL[999]
192 J P[1:Home] 100% FINE

```

A_BP_A_SUB (subprogram):

```

1 IF (R[120]=2 OR R[120]=4 OR R[120]=6) THEN
2 R[100]=R[100]+1
3 ENDF
4
5 !ACQUIRE 3D MAP
6 LBL[100]
7 !wait robot out of FOV
8 WAIT R[117]=1
9 CALL BINPICK_ACQUIRE3DMAP("3D Area Sensor"='FIXED')
10
11 !SEARCH
12 LBL[200]
13 CALL BINPICK_SEARCH("Parts List ID"=1,"SEARCH VP ID"=R[100],"Status R"=1
14 13)
15 IF R[113]=0,JMP LBL[300]
16 !SEARCH Fail
17 !sys status
18 R[110]=1
19 !SubPrg complete
20 R[112]=1
21 END
22
23 !POP
24 LBL[300]
25 CALL BINPICK_POP("Parts List ID"=1,"Status R"=114,"Model ID R"=115)
26 IF R[114]=0,JMP LBL[400]
27 !else = POP Fail
28 JMP LBL[100]
29
30 !GETPICKPOS
31 LBL[400]
32 CALL BINPICK_GETPICKPOS("Parts List ID"=1,"PICK Pos ID"=R[100],"Status R
33 "=116,"PICK Pos PR"=50,"IA Offset PR"=51,"APP Pos PR"=52)
34 IF R[116]=0,JMP LBL[410]
35 !else = GETPICKPOS Fail
36 CALL BINPICK_SETSTAT("Parts List ID"=1,"Part Status"=22)
37 JMP LBL[300]
38
39 !GETPICKPOS Success
40 LBL[410]
41 R[112]=1

```

Per il corretto funzionamento del processo, sono stati impostati i registri riportati in Tab. [4.1](#).

R[100]	ID componente da cercare.
R[110]	System status.
R[111]	Hold part (componente preso).
R[112]	Sottoprogramma completato.
R[113]	SEARCH status.
R[114]	POP status.
R[115]	ID popped.
R[116]	PICK status.
R[117]	Robot in FOV.
R[120]	Componenti piazzati.

Tabella 4.1: Registri utilizzati dalle funzioni macro nel programma main e subprogram.

Come anticipato, il main esegue le medesime operazioni del programma di base, fino alla presa del componente compresa di depart, a riga 84. Dopo averla eseguita, il robot si trova sopra i contenitori, con il componente stesso nelle griffe, in quell'istante viene avviato il subprogram (riga 93) ma il main non viene interrotto, prosegue andando alla posizione intermedia. Il subprogram attende che il robot non sia nel campo inquadrato dal sistema di visione, questo è rappresentato dal wait di riga 8 nel medesimo programma, il registro 117 viene modificato da un altro sottoprogramma chiamato *A_FOV_A*, che viene richiamato mezzo secondo dopo che il robot ha raggiunto la posizione P4, definita intermedia perché tra i contenitori e la shadow mask, il robot non è più nel FOV quindi il registro può essere modificato, facendo proseguire il subprogram. Questo come detto esegue un'acquisizione, cerca ed estrae dalla Parts List il componente con maggiore priorità, al termine pone il registro 112 pari ad 1, il main dopo aver eseguito il place dell'oggetto rimane in attesa che tale registro venga posto ad 1, in modo da poter poi proseguire con l'esecuzione, portandosi alla label 600 ossia ad un nuovo pick, dato che posizione ed orientazione del successivo componente sono già state ottenute grazie al subprogram.

Il ciclo completo di lavoro per comporre il kit prevede che questo venga completato aggiungendo un componente per volta nel seguente ordine: due pistoni, due bielle, due ruote dentate secondarie e due ruote dentate primarie. Di conseguenza, ogni due componenti aggiunti alla shadow mask, quindi al kit, deve cambiare il processo di visione per la ricerca del componente ed i parametri relativi alla Interference Avoidance visto che il nuovo componente non è nello stesso contenitore del precedente. Per questo motivo è stato utilizzato R[100], un registro che rappresenta il Vision Process ID nella SEARCH VP List ed il PICK Pos ID nella PICK Position List. Se posto pari ad 1, ad esempio, corrisponde al primo processo di visione nella SEARCH VP List (Fig. 4.10) ossia quello atto al riconoscimento dei pistoni, e nella PICK Position List (Fig. 4.11) corrisponde al primo elemento della lista, avente il contenitore dei pistoni come IASYS ed il medesimo

Vision Process. In questo modo se $R[100]$ è pari ad 1, i programmi eseguiranno un SEARCH dei pistoni, calcolandone posizione ed orientazione rispettando le condizioni per l'Interference Avoidance del corrispettivo contenitore. Tale registro è stato reso coincidente anche con l'ID del componente, ottenendo un'ulteriore uniformità per i numeri identificativi. Ad ogni componente piazzato, il registro $R[120]$ viene incrementato di un'unità ed ogni due componenti dello stesso tipo piazzati nella shadow mask, $R[100]$ viene anch'esso incrementato di uno, di conseguenza la ricerca dei componenti passa dai pistoni, alle bielle, alle ruote dentate secondarie ed infine alle ruote dentate primarie, di fatto componendo il kit nell'ordine desiderato. Questo giustifica l'ordine di definizione della SEARCH VP List e della PICK Position List descritte in precedenza. Il registro $R[100]$ viene utilizzato nelle macro BINPICK_SEARCH e BINPICK_GETPICKPOS, che necessitano delle informazioni appena citate. La condizione per il cambio di tipo di componente è, come detto, che siano stati depositati due componenti dello stesso tipo nella shadow mask, quindi il numero di componenti salvato nel registro $R[120]$ deve essere un multiplo di due, per questo il controllo è nel subprogram, nelle righe 1÷3, dove si considera se il medesimo registro in quel momento ha il valore di 2 4 o 6. Tale controllo viene eseguito nel subprogram e non nel main perché il primo esegue il processo di SEARCH in anticipo rispetto al deposito del componente, di fatto mentre il robot ha il secondo componente di un tipo nelle griffe, il subprogram esegue una scansione e cerca quello che sarà il primo componente del tipo successivo. Il registro 120 viene utilizzato anche per il controllo del place, in base al numero di componente questo ha una specifica posizione all'interno della shadow mask, di fatto si hanno otto differenti posizioni

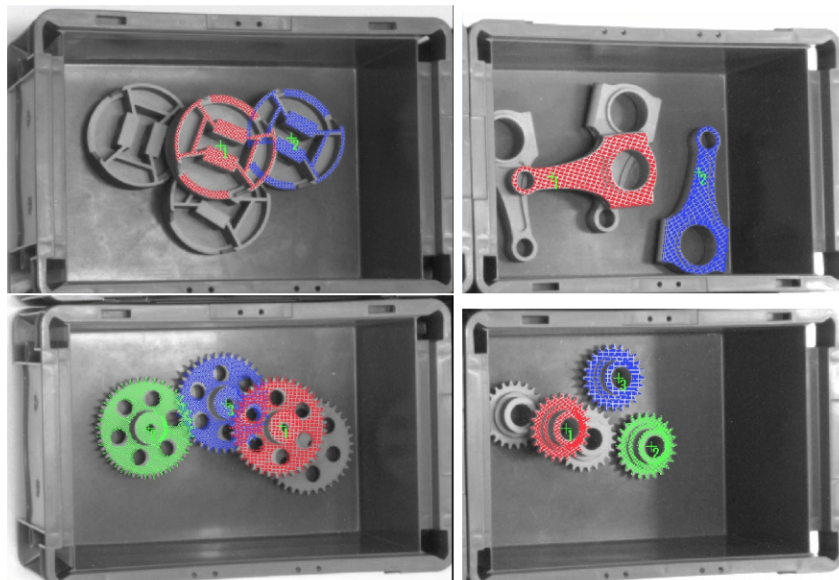


Figura 4.13: Risultati dei processi di visione sui componenti del kit.

di place, in base al componente, questi controlli sono eseguiti da riga 98 a riga 179. In aggiunta, se il robot ha nelle griffe l'ottavo ed ultimo componente, ossia la seconda ruota dentata primaria, non è necessario venga eseguito il subprogram, visto che implicherebbe una nuova acquisizione ed un nuovo SEARCH, di fatto non necessari visto che il kit è completo. Per questo, nel main a riga 92, appena prima della chiamata del subprogram, viene controllato se il componente in quel momento movimentato è l'ottavo, nel caso lo fosse il programma salta alla label 800, eseguendo il place della ruota dentata primaria per poi passare alla label 999 dove il robot viene portato alla posizione di Home ed il programma termina.

In Fig. 4.13 sono illustrati i risultati dei processi di visione per ogni componente utilizzato nel kitting.

Conclusioni

In conclusione, il sistema di visione testato risulta in grado di riconoscere componenti di dimensioni inferiori e complessità maggiore rispetto agli oggetti per il quale è stato progettato. Il kitting, come descritto, è composto da oggetti con geometrie assai più complesse e dimensioni notevolmente inferiori rispetto alle scatole standard utilizzate nei pallet. La composizione di tale kit, quindi il rilevamento dei quattro tipi di componenti, è stata possibile solo a seguito di numerosi test, che hanno permesso un fine tuning dei Vision Tools, in termini di definizione del modello e regolazione dei parametri. In questo modo, il 3DA si è rivelato robusto, in quanto non sensibile alle variazioni nell'illuminazione ambientale, e molto accurato, poiché ha permesso una presa dei componenti estremamente precisa, con un'ottima ripetibilità. La criticità della finitura superficiale del componente è invece caratteristica intrinseca della tecnologia impiegata nel sistema, quindi risulta una problematica non risolvibile. Il sistema di visione analizzato non è quindi adatto ad applicazioni in cui i componenti sono di materiale riflettente o trasparente, e sono di piccole dimensioni. Tuttavia, alla luce dei risultati ottenuti in termini di tempo per il riconoscimento ed accuratezza dello stesso, è possibile asserire che il Fanuc 3D Area Sensor può essere ampiamente impiegato in processi industriali di bin picking che prevedono la movimentazione di oggetti di piccole dimensioni, ad eccezione della minuteria (i.e. viti), rendendo tale sistema ancora più flessibile. I tempi impiegati per l'acquisizione 3D e per il rilevamento dei componenti, unitamente alla possibilità di eseguire lo stesso rilevamento mentre il robot è in movimento, rendono la cella robotizzata Fanuc adatta al processo industriale di kitting descritto in questo documento, tale cella può quindi essere utilizzata in un reale processo industriale.

Bibliografia

- [1] V. M. Manghisi, A. E. Uva, M. Fiorentino, V. Bevilacqua, G. F. Trotta, and G. Monno, “Real time rula assessment using kinect v2 sensor,” *Applied Ergonomics*, vol. 65, pp. 481–491, 2017.
- [2] J. Geng, “Structured-light 3d surface imaging: a tutorial,” *Adv. Opt. Photon.*, vol. 3, pp. 128–160, Jun 2011.
- [3] A. K. Gary Bradski, *Learning OpenCV*, pp. 415–418. O’Reilly Media, Inc, 09 2008.
- [4] Fanuc Corporation, “FANUC Robot series, R30iB CONTROLLER, *iRVision* Bin Picking Application OPERATOR’S MANUAL, B-83914EN-6/02,” 10th June 2020.
- [5] Fanuc Corporation, “FANUC Robot series, R30iB CONTROLLER, *iRVision* OPERATOR’S MANUAL (Reference), B-83304EN/04,” 8th March 2021.