



**UNIVERSITÀ DEGLI STUDI DI ROMA  
TOR VERGATA**

**FACOLTÀ DI INGEGNERIA**

**CORSO DI LAUREA IN INGEGNERIA  
DELL'AUTOMAZIONE**

**A.A. 2008/2009**

**Tesi di Laurea**

**Sistema di visione per la guida di un manipolatore**

**RELATORE**

Prof. Francesco Martinelli

**CANDIDATO**

Raffaele Rossi

# Indice

<b>Introduzione</b>	<b>1</b>
<b>1 L'immagine processing</b>	<b>3</b>
1.1 Le immagini digitali . . . . .	3
1.2 Sogliaatura . . . . .	4
1.3 Filtro anti-rumore . . . . .	5
1.4 Segmentazione . . . . .	6
<b>2 Il processing</b>	<b>8</b>
2.1 L'edge detection . . . . .	8
2.1.1 Il metodo del gradiente . . . . .	8
2.1.2 Il metodo di Canny . . . . .	11
2.2 I momenti . . . . .	12
2.3 Dall'immagine allo spazio di lavoro . . . . .	14
<b>3 La calibrazione</b>	<b>15</b>
3.1 L'idea . . . . .	15
3.2 La trasformata di Hough . . . . .	17
3.2.1 La retta . . . . .	17
3.2.2 L'algoritmo . . . . .	17

---

3.3	Usò della trasformata . . . . .	19
<b>4</b>	<b>Il robot</b>	<b>22</b>
4.1	Gli aspetti generali . . . . .	22
4.2	Cinematica diretta . . . . .	23
4.3	Cinematica inversa . . . . .	25
<b>5</b>	<b>La realizzazione</b>	<b>28</b>
5.1	Una panoramica sul funzionamento . . . . .	28
5.2	La libreria Video4Linux . . . . .	30
5.3	La libreria Qt . . . . .	31
5.4	La comunicazione TCP/IP . . . . .	32
<b>6</b>	<b>Conclusioni e sviluppi futuri</b>	<b>34</b>
	<b>Elenco delle figure</b>	<b>36</b>
	<b>Bibliografia</b>	<b>36</b>

# Introduzione

Da decine di anni i manipolatori occupano una posizione centrale all'interno di impianti industriali, sia per quanto riguarda la produzione in grande scala, sia per quanto riguarda l'espletamento di compiti rischiosi, inagibili o alienanti per l'uomo. Durante questo periodo di attività, i manipolatori hanno avuto sviluppi tecnologici che hanno permesso di operare con accuratezze sempre maggiori, su oggetti sempre più grandi e pesanti o tali da conferire maggior destrezza nella movimentazione dell'organo terminale (*effettore*). Si possono, infatti, trovare manipolatori in applicazioni quali saldature di componenti elettronici o strutture metalliche, verniciature, packaging, *etc...*

Nonostante l'impiego dei manipolatori, esistono ancora mansioni in cui l'uomo svolge lavori ripetitivi come, ad esempio, il packaging o il controllo di qualità.

Contemporaneamente, l'aumento della potenza di calcolo dei processori moderni, il miglioramento delle prestazioni delle telecamere e l'evoluzione di algoritmi di image processing, hanno reso possibile l'estrapolazione di quelle informazioni contenute nelle immagini, quali, ad esempio, posizione, orientamento, forma e dimensione di oggetti, utilizzabili per assolvere in modo automatico i suddetti compiti.

Il lavoro svolto in questa tesi mostra proprio come l'estrapolazione di tali informazioni, a partire da fotografie del *piano di lavoro* di un manipolatore, sia utilizzabile per guidare lo stesso affinché possa espletare il compito richiesto.

A tale scopo nel capitolo 1 verranno presentate tecniche generali di image processing, in modo da comprendere più chiaramente quali siano le possibilità offerte da tale strumento. In particolare, saranno discussi tre esempi di image processing: sogliatura, filtro anti-rumore ed un accenno alla trasformata di Hough.

Nel capitolo 2 verrà illustrata la sequenza di elaborazioni utilizzata per individuare la posizione e l'orientamento dell'oggetto. Poiché tutto è basato sul riconoscimento dei bordi dell'oggetto, verranno, dapprima, comparate due diverse tecniche di *edge detection* (gradiente e filtro di Canny), dopodiché si mostrerà come, attraverso il calcolo dei *momenti*, sia semplice ricavare posizione ed orientamento della *regione d'interesse*. Verrà, inoltre, introdotto il problema della calibrazione del sistema di visione, trattato nel capitolo 3. In quest'ultimo, infatti, si porrà l'attenzione sul problema di calcolare, tramite una procedura automatica, il valore di quei parametri necessari per riportare le informazioni di posizione ed orientamento dall'immagine allo *spazio operativo* del manipolatore, terminando così la parte riguardante l'image processing. Nel capitolo 4 si discuterà rapidamente del manipolatore e dei modelli matematici utilizzati per calcolare la cinematica diretta ed inversa.

Avendo, a questo punto, discusso tutti gli elementi che costituiscono il sistema, nel capitolo 5 verrà affrontato il problema della sua realizzazione, cioè della scrittura di un software in grado di catturare immagini da una webcam tramite la libreria Video4Linux, di visualizzare a schermo sia l'immagine catturata che i risultati dei processamenti svolti ed, infine, di comunicare via TCP/IP con il calcolatore dedicato al controllo del manipolatore. Quest'ultima caratteristica, inoltre, permette di espandere i campi di applicazione del sistema. Tale argomento sarà trattato nel capitolo 6 insieme agli sviluppi futuri.

# Capitolo 1

## L'immagine processing

*In questo capitolo verranno discusse alcune tecniche di image processing in modo da comprenderne maggiormente le possibilità offerte.*

### 1.1 Le immagini digitali

Le immagini contenute in un computer sono rappresentate da una matrice, i cui elementi (detti *pixel*) codificano l'intensità luminosa (nel caso di immagini monocromatiche dette anche *in scala di grigi*) di un punto nello spazio. Pertanto, denominando con  $x$  ed  $y$  le coordinate di un pixel, si può rappresentare l'immagine come una funzione quantizzata in due variabili  $F(x, y)$ . Inoltre, nel caso in cui l'immagine rappresentata sia a colori è possibile codificare il colore del singolo pixel su 3 livelli<sup>1</sup> (rosso, verde e blu) ed associare a ciascuno di essi una funzione quantizzata in  $x$  ed  $y$  analogamente a quanto fatto per le immagini in scala di grigi, cioè:

$$\begin{cases} F_R(x, y) \\ F_G(x, y) \\ F_B(x, y) \end{cases}$$

---

<sup>1</sup>Esistono ulteriori codifiche per le immagini a colori, tuttavia verranno prese in esame solo immagini con codifica RGB.

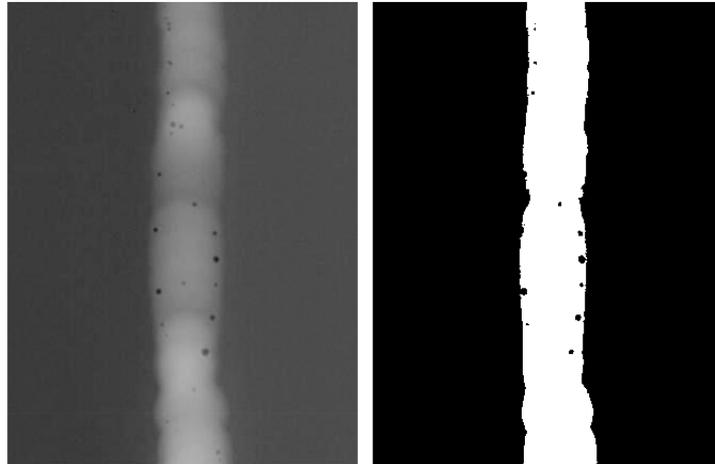


Figura 1.1: Imperfezioni in una saldatura evidenziate tramite sogliatura

Immagini molto particolari sono le cosiddette *immagini binarizzate*, cioè immagini per cui la funzione  $F(x, y)$  può assumere solo valori nell'insieme  $\{0,1\}$ .

## 1.2 Sogliatura

Una particolare applicazione delle immagini binarizzate permette di isolare le *regioni di interesse*, cioè il luogo di quei punti che rispettano una data relazione.

Fra le più semplici relazioni compare sicuramente la disugaglianza, pertanto si può introdurre una prima tecnica di image processing detta *sogliatura*.

Data un'immagine  $F(x, y)$  ed una soglia  $t$ , l'immagine  $G(x, y)$  ottenuta tramite sogliatura di  $F(x, y)$  è definita come:

$$G(x, y) = \begin{cases} 1 & \text{se } F(x, y) > t \\ 0 & \text{se } F(x, y) \leq t \end{cases}$$

Un primo utile esempio della sogliatura mostra come si possano mettere in risalto delle imperfezioni in una saldatura, partendo da un'immagine ottenuta con una telecamera a raggi X (vedi figura 1.1).

### 1.3 Filtro anti-rumore

Una delle prime esigenze a cui l'immagine processing tenta di porre rimedio, è la riduzione del rumore presente in un'immagine. Come è facile intuire, tale rumore è causato dai sensori che costituiscono le moderne fotocamere elettroniche e spesso è facilmente visibile ad occhio nudo.

Sfruttando la caratteristica di essere a media nulla, per cercare di eliminare il rumore dall'immagine, si esegue una convoluzione fra la funzione  $F(x, y)$  associata all'immagine ed una maschera  $M(x, y)$  associata ad una matrice quadrata  $M$ , di ordine  $n \times n$ , in modo da mediare ogni pixel di coordinate  $(x, y)$  con tutti i pixel appartenenti ad una regione centrata nelle coordinate  $(x, y)$  e di larghezza  $n \times n$ . In termini matematici, denominando con  $G(x, y)$  l'immagine dopo l'applicazione del filtro, risulta

$$G(x, y) = \sum_{i=-\frac{n-1}{2}}^{\frac{n-1}{2}} \sum_{j=-\frac{n-1}{2}}^{\frac{n-1}{2}} F(x+i, y+j)M(i, j)$$

La più semplice funzione  $M(x, y)$  utilizzabile è tale da eseguire la media aritmetica di tutti i pixel all'interno della suddetta regione di dimensioni  $n \times n$  e pertanto

$$M(x, y) = \frac{1}{n^2} \quad \forall(x, y)$$

Tuttavia, la maschera utilizzata nella maggioranza dei casi è la funzione gaussiana in due variabili, centrata nelle coordinate  $(n_0, n_0)$ , dove  $n_0$  è tale che  $n = 2n_0 + 1$ . Pertanto si può scrivere:

$$M(x, y) = e^{-\frac{(x-n_0)^2+(y-n_0)^2}{2\sigma^2}}$$

Regole pratiche suggeriscono di scegliere  $n$  come il più piccolo numero dispari superiore o uguale a  $6\sigma$ .



Figura 1.2: Un'immagine affetta da molto rumore ed il risultato dell'applicazione di un filtro con maschera gaussiana

La figura 1.2 mostra il risultato dell'applicazione del filtro anti-rumore, con maschera gaussiana, ad un'immagine affetta da molto rumore.

## 1.4 Segmentazione

Benché non appartenga alla classe delle elaborazioni che costituiscono l'immagine processing, la segmentazione è lo strumento centrale per quanto riguarda l'estrapolazione delle informazioni contenute in un'immagine.

Un esempio importante è, indubbiamente, costituito dalla trasformata di Hough, grazie alla quale è possibile individuare all'interno di un'immagine (o più precisamente all'interno di una regione d'interesse) le principali rette. La figura 1.3 mostra una scacchiera e le sue principali rette individuate tramite l'ausilio della trasformata di Hough.

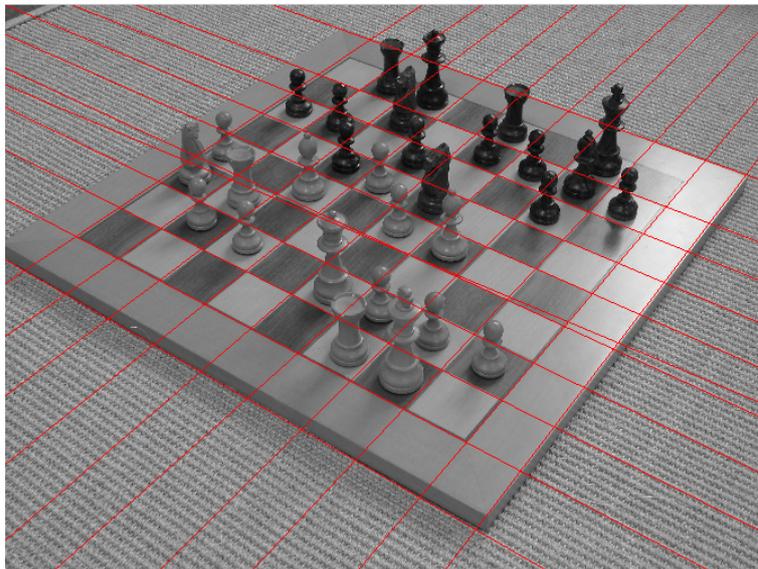


Figura 1.3: Le principali rette in una scacchiera

# Capitolo 2

## Il processing

*In questo capitolo verranno discusse le tecniche di image processing utilizzate per individuare l'oggetto (edge-detection), trovarne posizione ed orientamento (tramite i momenti) e verrà introdotto il problema della calibrazione del sistema.*

### 2.1 L'edge detection

Il primo problema in cui si incorre è individuare (o più precisamente delineare) un oggetto all'interno di un'immagine. A tale scopo si concentra l'attenzione sui bordi che costituiscono l'oggetto a cui si è interessati. Quindi, delineare l'oggetto si traduce in cercarne i bordi.

Supponendo che l'immagine da processare sia in scala di grigi, si possono individuare i bordi dell'oggetto interessato semplicemente cercando quei punti la cui intensità si distacca “notevolmente” dallo sfondo. Questo distacco si può valutare tramite il modulo del vettore gradiente.

#### 2.1.1 Il metodo del gradiente

Data un'immagine rappresentata dalla funzione  $F(x, y)$  associata ad una matrice  $F$  di ordine  $w \times h$  e tenendo conto che la minima variazione  $\Delta x$  è pari ad 1, si può

calcolare il gradiente di  $F(x, y)$  in direzione  $x$  come:

$$F_x(x, y) = F(x + 1, y) - F(x, y)$$

Analogamente, in direzione  $y$  si ha:

$$F_y(x, y) = F(x, y + 1) - F(x, y)$$

È, ora, possibile scrivere in forma compatta il modulo del vettore gradiente, calcolato in un qualsiasi punto  $(x, y)$ , tramite la matrice  $M$  associata alla funzione:

$$M(x, y) = \begin{cases} \sqrt{F_x(x, y)^2 + F_y(x, y)^2} & x \in [1, w - 1] \quad e \quad y \in [1, h - 1] \\ 0 & x = w \quad oppure \quad y = h \end{cases}$$

Avendo a disposizione le componenti in direzione  $x$  ed  $y$  del vettore gradiente, si può calcolare anche la direzione dello stesso, esprimibile dalla funzione:

$$\alpha(x, y) = \begin{cases} \text{atan2}(F_y(x, y), F_x(x, y)) & x \in [1, w - 1] \quad e \quad y \in [1, h - 1] \\ 0 & x = w \quad oppure \quad y = h \end{cases}$$

La figura 2.1 mostra un'immagine in scala di grigi che immortalava alcune monete e la rappresentazione grafica della relativa matrice  $M$ . Si noti l'inspessimento dei bordi delle monete, dovuto alla duplice variazione dell'intensità nel passaggio dallo sfondo al bordo della moneta e successivamente dal bordo della moneta alla regione interna.

Poiché si vuole concentrare l'attenzione solo sui bordi di un oggetto, tenendo conto di quanto detto nel capitolo 1.2, si può applicare una sogliatura alla matrice  $M$  per isolare quei punti in cui si ha una forte variazione dell'intensità ed ottenere, così, un'immagine binarizzata in cui si ha 1 in corrispondenza del bordo. Applicando, ad esempio, alla matrice  $M$  rappresentata in figura 2.1, una sogliatura scegliendo  $t = 0.25 \max(M(x, y))$ , si ottiene l'immagine in figura 2.2.

Si noti come l'inspessimento dei bordi venga messo in risalto dalla sogliatura.

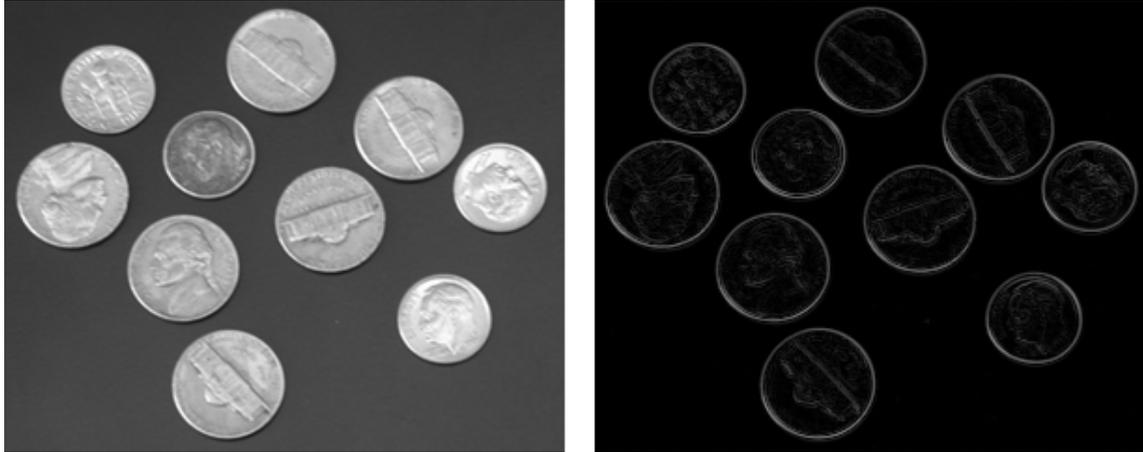


Figura 2.1: La fotografia di alcune monete e la rappresentazione grafica del modulo dei vettori gradiente

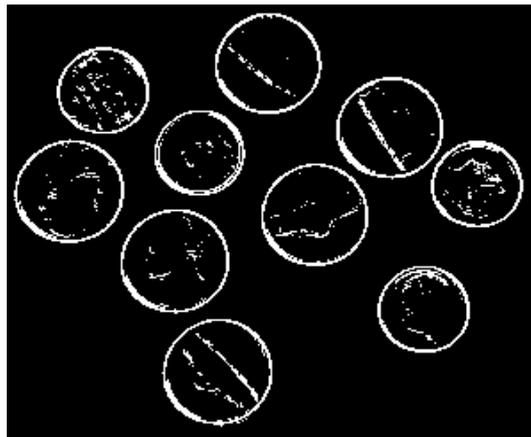


Figura 2.2: Applicazione della sogliatura alla figura 2.1

### 2.1.2 Il metodo di Canny

Come è facile intuire, la scelta della soglia  $t$  vista in precedenza è molto delicata. Infatti, valori troppo elevati di  $t$  potrebbero escludere punti appartenenti al bordo (*falsi negativi*), mentre valori troppo bassi potrebbero introdurre punti non appartenenti al bordo (*falsi positivi*). Inoltre, si potrebbe definire il bordo come una regione di larghezza infinitesima (almeno nella realtà), pertanto si vorrebbe limitare l'inspessimento dovuto al calcolo del gradiente e l'enfatizzazione introdotta dalla sogliatura. Il filtro di Canny pone rimedio ad entrambi questi problemi.

Per quanto riguarda l'inspessimento, Canny introduce, prima ancora di applicare la sogliatura alla matrice  $M$ , uno stadio di *assottigliamento* basato su una tecnica denominata *non-maxima suppression*. L'idea alla base di tale tecnica sfrutta la caratteristica del gradiente di essere ortogonale al bordo. Spostandosi in direzione del gradiente, infatti, si sopprimono (o ignorano) tutti quei punti il cui modulo è strettamente minore del modulo di uno dei due punti ad esso adiacenti, andando così a costruire una nuova matrice  $F_M$  su cui successivamente applicare la sogliatura. Tale sogliatura, però, differisce da quella vista in precedenza, in quanto si basa sull'utilizzo di due soglie  $t_H$  e  $t_L$ <sup>1</sup>, cercando, in questo modo, di contenere il problema dei falsi positivi e falsi negativi. Sfruttando le due soglie appena descritte, si definiscono *punti forti* tutti quei punti il cui modulo è maggiore di  $t_H$ , mentre si definiscono, *punti deboli* tutti quei punti, il cui modulo è maggiore di  $t_L$ . È immediato osservare che il numero di punti forti è inferiore al numero dei punti deboli ed, inoltre, tutti i punti forti possono essere istantaneamente segnati come punti di bordo. L'ultimo stadio del filtro di Canny ha l'obiettivo di ricostruire il bordo degli oggetti, considerando come punti di bordo tutti i punti deboli *connessi* a punti forti, ovvero ripristinare tutti i

---

<sup>1</sup>regole pratiche suggeriscono di scegliere  $t_H$  pari a 2 o 3 volte  $t_L$

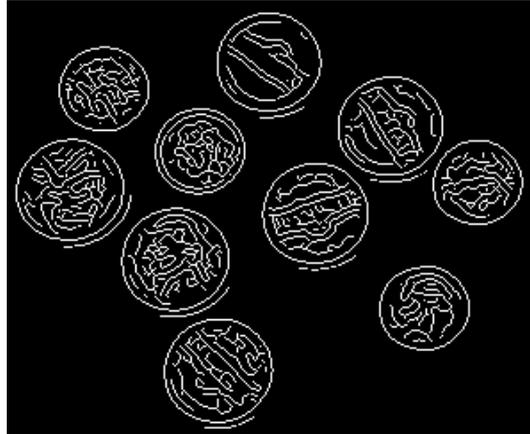


Figura 2.3: Applicazione del filtro di Canny alla fotografia in figura 2.1

falsi negativi. Si ha, pertanto, l'esigenza di scegliere un criterio di *connessione*, cioè di una relazione che stabilisca se due punti siano connessi. Una semplice scelta per il criterio di connessione è la seguente:

*Dati due punti  $P_1 = (x_1, y_1)$  e  $P_2(x_2, y_2)$ , essi si definiscono connessi se  $|x_1 - x_2| \leq 1$  oppure  $|y_1 - y_2| \leq 1$ , ovvero se  $P_2$  appartiene alla regione  $3 \times 3$  centrata in  $P_1$ .*

Si possono, tuttavia, definire criteri di connessione basati sui diversi tipi di cammino descritti nella *teoria dei grafi*.

L'immagine 2.3 mostra l'applicazione del filtro di Canny all'immagine in scala di grigi di figura 2.1.

## 2.2 I momenti

Una volta delineata la regione d'interesse, bisogna valutarne posizione ed orientamento. A tale scopo si prestano facilmente i *momenti*.

Data un'immagine  $F(x, y)$  in scala di grigi, si definisce momento  $m_{i,j}$  il valore:

$$\sum_X \sum_Y F(x, y) x^i y^j$$

Nel caso in cui l'immagine  $F(x, y)$  sia binarizzata, il momento  $m_{i,j}$  si riduce ad essere:

$$\sum_X \sum_Y x^i y^j$$

Si definisce, inoltre, *ordine* del momento  $m_{i,j}$  il valore  $i + j$ .

È immediato osservare che, il momento  $m_{0,0}$  è pari al numero di pixel ad 1 nell'immagine binarizzata (o regione d'interesse in base a quanto detto nel capitolo 1.2) ed, infatti, viene denominato *area della regione d'interesse*.

Un semplice metodo per calcolare le coordinate  $(x_c, y_c)$ , del centro della regione d'interesse (o *centroide*), è la media aritmetica dei contributi lungo gli assi  $x$  ed  $y$  di ciascuno dei punti d'interesse. In particolare, si può scrivere:

$$x_c = \frac{m_{1,0}}{m_{0,0}} \quad , \quad y_c = \frac{m_{0,1}}{m_{0,0}}$$

Avendo, ora, a disposizione le coordinate del centroide, è possibile definire il *momento centrale* (o *invariante alla traslazione*)  $\mu_{i,j}$  come:

$$\sum_X \sum_Y (x - x_c)^i (y - y_c)^j$$

I momenti centrali del second'ordine permettono di calcolare l'angolo che l'*asse maggiore* della regione forma con l'asse  $x$  dell'immagine, tramite la formula:

$$\alpha = \frac{1}{2} \arctan \left( \frac{2\mu_{1,1}}{\mu_{0,2} - \mu_{2,0}} \right)$$

Inoltre, gli autovalori (e relativi autovettori) della matrice:

$$\begin{bmatrix} \mu_{0,2} & \mu_{1,1} \\ \mu_{1,1} & \mu_{2,0} \end{bmatrix}$$

forniscono alcune basilari informazioni in merito a forma della regione e alla direzione dei suoi assi principali (o, meglio, dell'ellisse approssimante). Ulteriori informazioni

possono essere ricavate tramite la valutazione dei *momenti invarianti normalizzati*, cioè di un insieme particolare di momenti il cui valore è invariante a trasformazioni quali traslazione, rotazione, scalatura e ribaltamento. Tuttavia, i momenti invarianti normalizzati non sono oggetto della presente trattazione.

## 2.3 Dall'immagine allo spazio di lavoro

Quanto visto finora permette di calcolare posizione ed orientamento di una regione rispetto al sistema di riferimento dell'immagine, la cui origine è posizionata nel pixel in alto a sinistra. Per poter, però, trasformare tali informazioni nella posizione e nell'orientamento rispetto al sistema di riferimento in cui opera il robot, occorre valutare un certo insieme di parametri necessari ad effettuare le dovute traslazioni, rotazioni e scalature. Tali parametri dipendono dalla posizione e dall'orientamento della telecamera rispetto al sistema di riferimento in cui opera il manipolatore.

Per ricavare questi parametri, si è sviluppato un algoritmo di calibrazione automatica, così da aumentare la flessibilità complessiva del sistema.

# Capitolo 3

## La calibrazione

*In questo capitolo verrà esposto l'algoritmo di calibrazione, tramite il quale valutare automaticamente i parametri necessari a trasformare la posizione e l'orientamento della regione d'interesse nella posizione e nell'orientamento dell'oggetto rispetto al sistema di riferimento in cui opera il robot. A tale scopo si è fatto uso della trasformata di Hough, di cui verrà data spiegazione.*

### 3.1 L'idea

Come detto nel capitolo 2, i parametri per trasformare posizione ed orientamento dall'immagine allo *spazio operativo*, cioè rispetto al sistema di riferimento in cui opera il robot, dipendono dalla posizione e dall'orientamento della telecamera nello spazio operativo stesso. Si potrebbe, quindi, pensare di ricavare tali informazioni da una o più immagini. Tuttavia, si è scelto di tracciare una regione quadrata all'interno dello spazio operativo, denominata *piano di lavoro*, e di riportare posizione ed orientamento rispetto ad uno dei suoi quattro vertici, di cui si conoscono le coordinate nello spazio operativo. In questo modo, per quanto riguarda la posizione, sarà sufficiente calcolare la distanza del centroide rispetto ai due lati del piano di lavoro passanti per il vertice che si sta considerando e, successivamente, traslare il risultato ottenuto in base alle coordinate del vertice stesso. Mentre, per quanto riguarda l'orientamento,

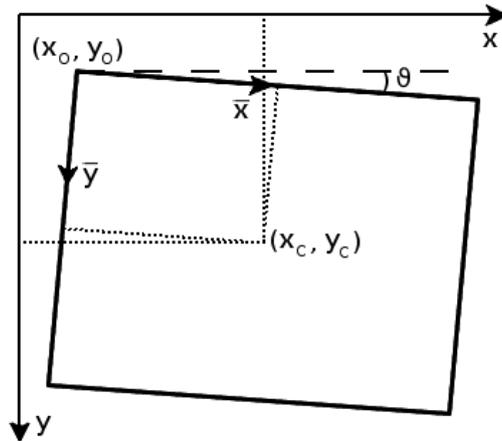


Figura 3.1: Posizione ed orientamento del piano di lavoro in un'immagine

sarà sufficiente sommare all'angolo che esprime l'orientamento della regione d'interesse, l'angolo che il piano di lavoro assume nell'immagine<sup>1</sup> (si faccia riferimento alla figura 3.1). Un ulteriore parametro necessario ad eseguire la trasformazione descritta in precedenza, è il rapporto che sussiste fra pixel e millimetri, dipendente dall'altezza a cui viene fissata la telecamera. Conoscendo, pertanto, le coordinate nello spazio operativo dei quattro vertici del piano di lavoro, le coordinate di questi ultimi nell'immagine e l'orientamento che il piano di lavoro assume nell'immagine, è possibile calcolare automaticamente tali valori<sup>2</sup>.

L'elemento fondamentale, quindi, è il quadrato che delimita il piano di lavoro. Per riconoscere quest'ultimo, si tenta di individuare le rette passanti per ciascuno dei quattro lati e, successivamente, di estrapolarne i vertici semplicemente calcolando i punti di intersezione delle stesse, prese a due a due ortogonali. Una volta note le quattro rette, sarà possibile, inoltre, calcolare l'orientamento che il piano di lavoro assume nell'immagine, utilizzando il coefficiente angolare di una di queste.

<sup>1</sup>più l'eventuale angolo che il piano di lavoro assume nello spazio operativo

<sup>2</sup>assumendo un legame lineare fra i valori espressi in pixel e il corrispettivo in millimetri, ovvero avendo preventivamente verificato gli effetti delle distorsioni introdotte dalla telecamera

## 3.2 La trasformata di Hough

Per realizzare quanto finora descritto, si deve disporre di un metodo tramite il quale individuare una o più rette in un'immagine. Questo compito viene assolto dalla trasformata di Hough.

### 3.2.1 La retta

Prima ancora di descrivere la trasformata di Hough, si ha la necessità di rappresentare univocamente una retta tramite l'ausilio di parametri caratteristici. È immediato pensare che il modo più naturale di eseguire tale rappresentazione sia tramite la coppia coefficiente angolare ed ordinata all'ascissa  $(m, q)$  dell'equazione cardinale della retta  $y = mx + q$ . Al contempo, però, ci si rende subito conto che per le rette della forma  $x = x_0$  o, comunque, tali da avere coefficiente angolare molto elevato, tale rappresentazione è di scarsa praticità.

Si vorrebbe rappresentare univocamente una retta utilizzando dei parametri di valore limitato, così da non introdurre errori dovuti alla digitalizzazione e che, magari, garantiscano anche continuità nella rappresentazione. Guardando la figura 3.2 si nota facilmente come una retta sia univocamente rappresentabile tramite la coppia  $(r, \theta)$  che caratterizza il vettore avente origine nel punto  $(0, 0)$  ed ortogonale alla retta stessa. Si noti, inoltre, che una retta verticale è rappresentata dalla coppia  $(x_0, 0)$ <sup>3</sup>.

### 3.2.2 L'algoritmo

Facendo riferimento alla definizione di regione di interesse data nel capitolo 1.2, si potrebbe tentare, come primo approccio al problema di individuare le principali rette presenti in un'immagine, di calcolare per ogni coppia di pixel, appartenenti alla

---

<sup>3</sup>oppure  $(-x_0, \pi)$  nel caso in cui  $x_0 < 0$

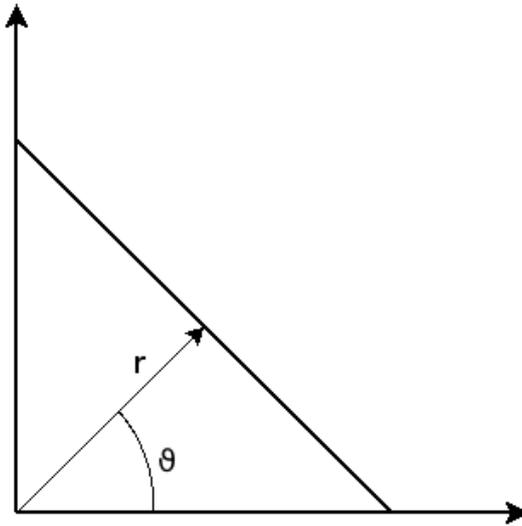


Figura 3.2: Rappresentazione vettoriale di una retta nel piano

regione di interesse, i parametri  $(r, \theta)$  dell'ipotetica retta passante per tali punti. Supponendo che due pixel appartengano alla stessa retta, essi forniranno una coppia  $(r_0, \theta_0)$  e con loro tutte le altre coppie di punti appartenenti alla medesima retta. È facile intuire, quindi, che le principali rette sono individuabili ricercando le coppie  $(r, \theta)$  di accumulo. Nonostante la sua semplicità ed efficacia, questa prima tecnica non viene utilizzata in quanto computazionalmente onerosa. Infatti, supponendo che la regione di interesse sia costituita da  $n$  punti, si devono effettuare  $n(n - 1) = o(n^2)$  operazioni.

La trasformata di Hough permette di giungere allo stesso risultato con un algoritmo più efficiente. Per ogni punto  $(x, y)$  del piano cartesiano esiste un fascio proprio di rette ottenibile variando i parametri  $(m, q)$  dell'equazione cardinale della retta<sup>4</sup>. Se due punti giacciono sulla stessa retta, delle infinite rette appartenenti a ciascuno dei due fasci propri, ne esisterà una, comune ad entrambi i fasci, i cui parametri  $(m, q)$  sono proprio i parametri della retta passante per tali punti. Riprendendo la

---

<sup>4</sup>A tale fascio di rette va aggiunta la retta  $x = x_0$

discussione fatta in precedenza, per una maggiore praticità si rappresenta la retta tramite i parametri  $(r, \theta)$ . Con l'ausilio di tale rappresentazione, tenendo conto che  $r = q \sin(\theta)$  e che  $m = -\tan(\theta)^{-1}$  si può esprimere il fascio di rette passante per il punto  $(x, y)$  tramite l'equazione in forma polare

$$r = y \sin(\theta) + x \cos(\theta)$$

Variando  $\theta$  nell'intervallo  $[0, 2\pi]^5$ , si può tracciare la sinusoidale (nel piano  $(r, \theta)$ ) che esprime graficamente la trasformata di Hough del punto  $(x, y)$ . Il punto di intersezione fra due di queste sinusoidi è la coppia  $(r_0, \theta_0)$  che rappresenta la retta passante per i due punti  $(x, y)$  di cui le sinusoidi sono le trasformate (a titolo di esempio si veda la figura 3.3). Si noti, comunque, che l'equazione del fascio di rette in forma polare fornisce anche valori negativi per  $r$ , in tal caso si può considerare il valore assoluto di  $r$ , purché si aggiunga un termine  $\pi$  a  $\theta$ . Come per il caso precedente, i punti  $(r, \theta)$  di accumulazione rappresentano le candidate rette presenti nell'immagine. Il costo computazionale di questo algoritmo è  $o(n)$ .

### 3.3 Uso della trasformata

Avendo, a questo punto, a disposizione le rette sulle quali giacciono i lati del piano di lavoro, è possibile calcolarne le coordinate dei vertici e l'orientamento. Per far ciò si descrivono, innanzitutto, le quattro rette con le coppie  $(r_1, \theta_1), (r_2, \theta_2), (r_3, \theta_3), (r_4, \theta_4)$ , in modo tale da rispettare le condizioni

$$r_1 \leq r_2 \leq r_3 \leq r_4$$

Così facendo, il vertice in alto a sinistra (o, meglio, quello più vicino all'origine dell'immagine) corrisponderà all'intersezione fra le rette  $(r_1, \theta_1)$  ed  $(r_2, \theta_2)$ . Per ga-

---

<sup>5</sup>o un intervallo minore se desiderato

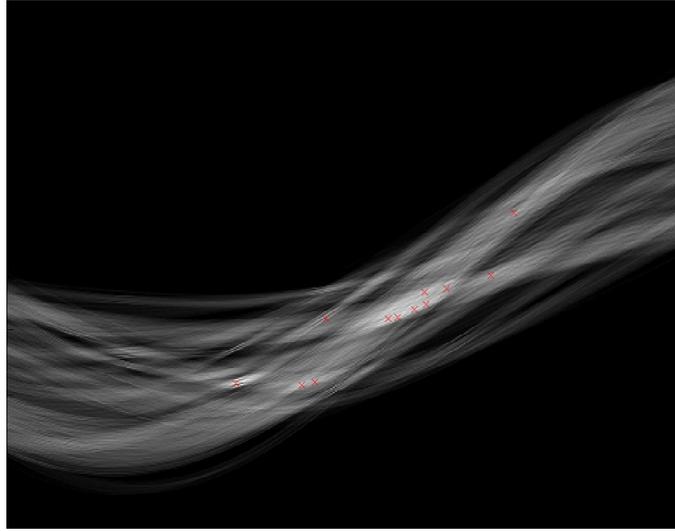


Figura 3.3: La trasformata di Hough graficamente

rantire, però, tale corrispondenza, è necessario supporre che la calibrazione serva a compensare errori di installazione non troppo elevati; è facile, infatti, intuire quali siano gli effetti nel caso in cui la telecamera venisse ruotata di  $90^\circ$ .

Conoscendo, inoltre, le coordinate dei quattro vertici (sia nello spazio operativo che nell'immagine), si possono calcolare i coefficienti  $\gamma_x$  e  $\gamma_y$  di proporzionalità fra pixel e millimetri. Per ogni segmento, infatti, denominando con  $\alpha$  l'angolo rispetto all'orizzontale (ovvero  $\theta_i + \frac{\pi}{2}$  con  $i = 1, 2, 3, 4$ ), con  $L_x$  e  $L_y$  le proiezioni del segmento sugli assi  $x$  ed  $y$ , rispettivamente e, infine, con  $L$  la lunghezza nello spazio operativo del segmento, si possono scrivere le relazioni:

$$\begin{cases} L_x \gamma_x = L \cos(\alpha) \\ L_y \gamma_y = L \sin(\alpha) \end{cases}$$

Per ridurre l'errore nella stima di tali parametri, si è calcolata la media aritmetica dei valori  $\gamma_x$  e  $\gamma_y$  ottenuti per ogni lato.

Da ultimo, per ricavare la posizione  $(x_p, y_p)$  dell'oggetto nello spazio operativo, si

devono calcolare le distanze del centroide della regione d'interesse dalle due rette  $(r_x, \theta_x)$  ed  $(r_y, \theta_y)$ , non necessariamente coincidenti con  $(r_1, \theta_1)$  ed  $(r_2, \theta_2)$ , si faccia nuovamente riferimento alla figura 3.1. Denominando con  $D_x$  e  $D_y$  tali distanze e con  $(x_0, y_0)$  le coordinate nello spazio operativo del vertice di riferimento, risulta:

$$\begin{cases} x_p = x_0 + \sqrt{(D_x \cos(\theta_x + \frac{\pi}{2})\gamma_x)^2 + (D_x \sin(\theta_x + \frac{\pi}{2})\gamma_y)^2} \\ y_p = y_0 + \sqrt{(D_y \cos(\theta_y + \frac{\pi}{2})\gamma_x)^2 + (D_y \sin(\theta_y + \frac{\pi}{2})\gamma_y)^2} \end{cases}$$

Inoltre, l'orientamento del piano di lavoro all'interno dell'immagine risulta essere

$$\theta_x + \frac{\pi}{2}$$

# Capitolo 4

## Il robot

*In questo capitolo verranno descritti gli aspetti generali di un manipolatore nonché il robot in dotazione (Scortec). Verrà presentato il modello matematico utilizzato per il calcolo della cinematica diretta (Denavit-Hartenberg) e la sua applicazione allo Scortec. Da ultimo sarà discusso il problema della cinematica inversa e il modello relativo allo Scortec.*

### 4.1 Gli aspetti generali

Un *robot manipolatore* è un sistema meccanico costituito da tre sottosistemi: la *struttura portante*, il *polso* e l'*effettore*. Il tipo di struttura portante, inoltre, permette di classificare il manipolatore e da essa dipende la forma geometrica dello *spazio operativo*, cioè il luogo dei punti raggiungibili dal robot. Esistono, ad esempio, robot *cartesiani*, *antropomorfi* o *SCARA*, come mostra la figura 4.1.

Lo Scortec è un robot antropomorfo, utilizzato prevalentemente per scopi didattici. L'effettore è una pinza capace di prendere oggetti di piccole dimensioni, in quanto ha un'apertura inferiore ai 10-12 centimetri. I motori elettrici sono dotati di encoder incrementali e motorizzazioni. Non avendo a disposizione specifiche tecniche, è stato necessario misurare manualmente la geometria dei membri, nonché valutare (attra-

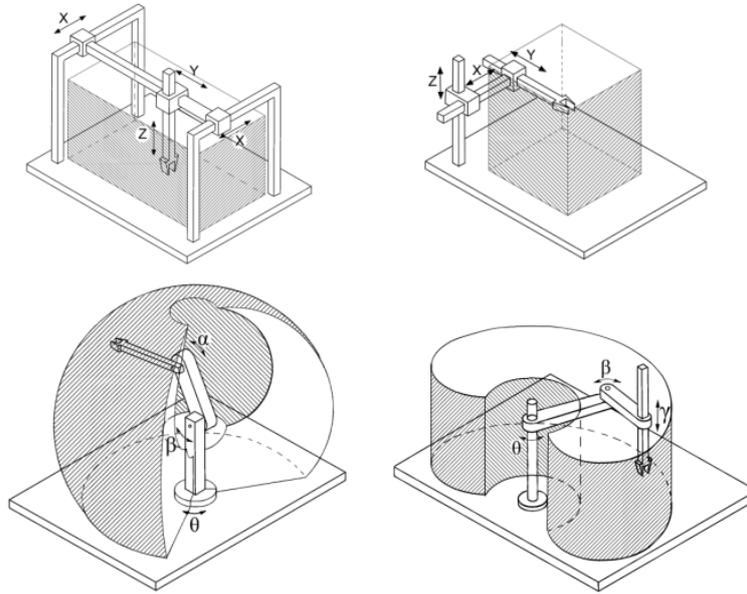


Figura 4.1: Due manipolatori cartesiani, un antropomorfo ed uno SCARA

verso misure successive) le proporzioni esistenti fra gli angoli di giunto e le misure provenienti dagli encoder.

## 4.2 Cinematica diretta

Quando si lavora con un manipolatore, il primo passo che si compie è, quasi sempre, il calcolo della cinematica diretta, cioè il calcolo della posizione e dell'orientamento dell'effettore a partire dai parametri di giunto<sup>1</sup>. A tale scopo si compongono le trasformazioni geometriche elementari (rotazione e traslazione) per ottenere un'unica, più complessa, trasformazione tramite la quale calcolare direttamente la posizione e l'orientamento di un giunto rispetto al giunto precedente. A sua volta, questo legame, è composto dalle suddette trasformazioni elementari. Affinché, però, sia possibile

<sup>1</sup>angoli nel caso di giunti rotoidali e distanze nel caso di giunti prismatici

comporre queste trasformazioni elementari, è necessario stabilire, per ogni giunto, un sistema di riferimento relativo al giunto stesso. La convenzione Denavit-Hartenberg stabilisce come posizionare gli assi di ciascuno di questi sistemi di riferimento, in funzione, anche, del sistema di riferimento relativo al giunto precedente. Una volta fissati tutti questi sistemi di riferimento, bisogna valutare per ciascuna coppia di giunti consecutivi i parametri  $a_i, d_i, \alpha_i, \theta_i$ , previsti dalla convenzione stessa. I parametri  $a_i$  e  $\alpha_i$  sono costanti, in quanto dipendono solamente dalla geometria del membro che connette i due giunti. In particolare, il termine  $a_i$  tiene conto della distanza fra gli assi dei due giunti, mentre il termine  $\alpha_i$  dell'angolo relativo fra gli stessi<sup>2</sup>. Gli ultimi due parametri, invece,  $d_i$  e  $\theta_i$  rappresentano proprio la posizione relativa fra i due giunti (distanza  $d_i$  nel caso di giunto prismatico o angolo  $\theta_i$  nel caso di giunto rotoidale). È importante osservare che, essendo ogni giunto ad un solo grado di libertà, uno dei due parametri è costantemente nullo. Il legame esistente fra posizione ed orientamento del giunto  $i$  rispetto al giunto  $i - 1$  è espresso dalla matrice:

$$A_i^{i-1} = \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i) \cos(\alpha_i) & \sin(\theta_i) \sin(\alpha_i) & a_i \cos(\theta_i) \\ \sin(\theta_i) & \cos(\theta_i) \cos(\alpha_i) & -\cos(\theta_i) \sin(\alpha_i) & a_i \sin(\theta_i) \\ 0 & \sin(\alpha_i) & \cos(\alpha_i) & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Applicando tale convenzione allo Scortec, e scegliendo i sistemi di riferimento posizionati come mostrato dalla figura 4.2, si ottiene la seguente tabella per i parametri DH<sup>3</sup>

<i>link</i>	<i>a</i>	<i>d</i>	$\alpha$	$\theta$
1	0	224	$\frac{\pi}{2}$	$\theta_0$
2	195	0	0	$\theta_1$
3	195	0	0	$\theta_2$
4	0	0	$\frac{\pi}{2}$	$\theta_3$
5	0	150	0	$\theta_4$

<sup>2</sup>valutato attorno all'asse  $x_i$  del sistema di riferimento del giunto  $i$

<sup>3</sup>le distanze sono espresse in millimetri

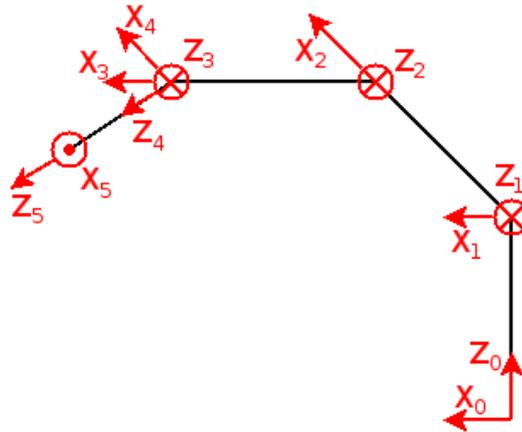


Figura 4.2: I sistemi di riferimenti scelti

### 4.3 Cinematica inversa

Nella movimentazione di un manipolatore, uno dei problemi di maggior rilievo è la soluzione della cinematica inversa, cioè del calcolo, a partire dalla posizione e dall'orientamento che si vuole far assumere all'effettore, dei parametri di giunto che portano il robot alla configurazione desiderata. A differenza della cinematica diretta, non esiste un metodo generale per risolvere tale problema e, infatti, la soluzione adottata dipende dalla struttura del manipolatore e viene, inoltre, complicata da eventuali gradi di libertà ridondanti.

In un manipolatore antropomorfo, come lo Scortec, si descrive la posizione e l'orientamento dall'effettore tramite, rispettivamente, la terna  $(x, y, z)$  e la coppia  $(\phi, \theta_4)$ , dove  $\phi$  è l'angolo che l'asse di avvicinamento dell'effettore ( $z_5$ ) forma col piano orizzontale. Per quanto riguarda  $\theta_0$ , risulta banalmente:

$$\theta_0 = \text{atan2}(y, x)$$

Più complicata è, invece, la determinazione degli angoli di giunto  $(\theta_1, \theta_2, \theta_3)$ . Per semplificare il problema, si effettua una traslazione, in modo da riferire il calcolo ri-

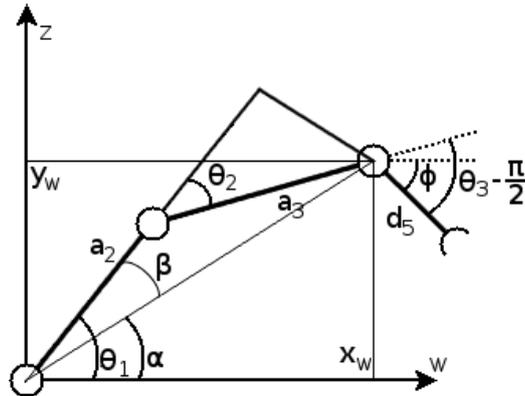


Figura 4.3: Il sistema di riferimento per la risoluzione della cinematica inversa.

spetto ad un nuovo sistema di riferimento la cui origine è posizionata sull'asse  $z_1$ , come mostra la figura 4.3. L'asse  $Z$  è l'asse parallelo a  $z_0^4$ , mentre  $W$  è l'asse perpendicolare a  $Z$  e tale da formare un angolo di ampiezza  $\theta_0$  con l'asse  $x$ .

Supponendo di aver ricavato gli angoli  $\theta_1$  e  $\theta_2$  che portano il polso nella posizione desiderata, si può banalmente ricavare l'angolo  $\theta_3$  osservando che

$$\phi = \theta_1 - \theta_2 + \theta_3 - \frac{\pi}{2}$$

Rimane, dunque, da ricavare i valori  $\theta_1$  e  $\theta_2$ . Essendo note le coordinate del punto  $(x_w, y_w)$ , in quanto noti i parametri  $d_5$  e  $\phi$ , si possono scrivere le equazioni:

$$\begin{aligned} x_w &= a_2 \cos(\theta_1) + a_3 \cos(\theta_1 - \theta_2) \\ y_w &= a_2 \sin(\theta_1) + a_3 \sin(\theta_1 - \theta_2) \end{aligned}$$

Dalle quali si può ricavare:

$$\cos(\theta_2) = \frac{x_w^2 + y_w^2 - a_2^2 - a_3^2}{2a_2a_3}$$

Invertibile solo se il punto  $(x, y, z)$  è raggiungibile con l'orientamento desiderato,

---

<sup>4</sup>Nel caso particolare della Scortec i due sono coincidenti

ovvero se è verificata la condizione.

$$\left| \frac{x_w^2 + y_w^2 - a_2^2 - a_3^2}{2a_2a_3} \right| \leq 1$$

Conoscendo, ora,  $\theta_2$ , è possibile ricavare  $\theta_1$ , come  $\alpha + \beta$ , dove:

$$\begin{aligned}\beta &= \text{atan2}(a_3 \sin(\theta_2), a_2 + a_3 \cos(\theta_2)) \\ \alpha &= \text{atan2}(y_w, x_w)\end{aligned}$$

# Capitolo 5

## La realizzazione

*Dopo aver visto tutte le componenti del sistema, in questo capitolo si discuterà della sua realizzazione. Verrà descritta brevemente la libreria Video4Linux che permette di catturare immagini e video da una webcam commerciale, la libreria Qt per la scrittura di programmi C++ con interfaccia grafica ed, infine, la comunicazione TCP/IP con il computer dedicato al controllo del manipolatore.*

### 5.1 Una panoramica sul funzionamento

Prima di scendere nei dettagli realizzativi, è importante capire più chiaramente le interazioni esistenti fra le varie componenti del sistema. A tale scopo si osservi lo schema a blocchi presentanto in figura 5.1.

Basandosi sull'immagine processing, il procedimento ha, ovviamente, inizio dalla cattura di un'immagine tramite la webcam. L'immagine digitalizzata viene trasmessa, su cavo USB, al calcolatore che si occupa di svolgere i processamenti necessari (Server). Il programma di elaborazione ha accesso all'immagine tramite le API della libreria Video4Linux e può, quindi, mostrarla a video. Sfruttando la programmazione concorrente, si possono lanciare diversi thread, ciascuno dedicato a svolgere un ben preciso compito. Ad occuparsi della presentazione a video dell'immagine catturata è il thread grafico, il cui scopo è la gestione dell'interfaccia grafica (GUI). Contemporaneamente

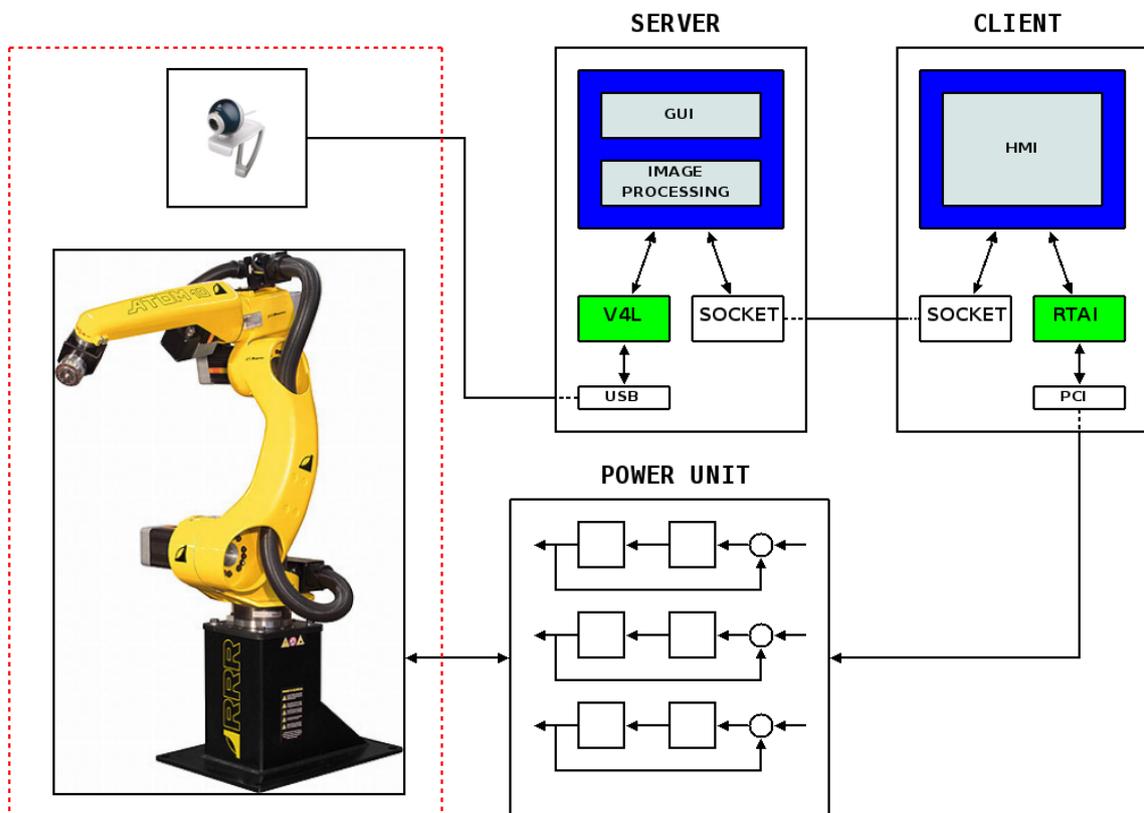


Figura 5.1: Schema a blocchi del sistema

te, un altro thread esegue gli algoritmi di image processing e segmentazione visti nel corso di questa trattazione. Una volta identificata la posizione e l'orientamento dell'oggetto interessato, il programma invia i comandi opportuni al client che controlla il manipolatore, tramite protocollo TCP/IP. Il client riceve la stringa binaria ed, una volta elaborata, invia gli ordini alla power unit che contiene al suo interno le catene di controllo per i singoli motori, realizzando così il comando impartito dal server.

## 5.2 La libreria Video4Linux

La libreria Video4Linux permette di realizzare le operazioni di input/output da e verso un dispositivo di cattura audio/video. Oltre ad essere una libreria, Video4Linux è una collezione di driver per diversi chipset di dispositivi. Le operazioni di lettura e scrittura avvengono (in linguaggio C) tramite istanze delle funzioni `'read()'` e `'write()'` della libreria standard `'unistd.h'`. Il compito svolto da Video4Linux è di gestire il dispositivo durante tutta la comunicazione, così da realizzare le operazioni di lettura e scrittura in modo del tutto trasparente al programmatore. Questo si traduce in una semplicità d'uso che permette all'utente di concentrarsi totalmente sulle elaborazioni. Inoltre, essendo pienamente in accordo con l'architettura generale dei sistemi operativi Unix, il dispositivo viene visto dal programmatore come un semplice file. Pertanto, i passi da compiere per integrare il dispositivo nel sistema sono:

**Apertura del dispositivo:** istanza della funzione `'open()'` sui device file del tipo `'/dev/video0'`.

**Configurazione del dispositivo:** successione di istanze della funzione `'ioctl()'` sul descrittore del dispositivo, ciascuna delle quali imposta il parametro specificato al valore desiderato.

**Lettura dell'immagine:** istanza della funzione `'read()'`, in cui la dimensione del buffer da leggere dipende dalla grandezza dell'immagine e dalla codifica utilizzata, ad esempio, 3 byte per pixel per la codifica RGB

**Chiusura del dispositivo:** alla conclusione del programma, un'istanza alla funzione `'close()'` permette di rilasciare opportunamente la risorsa.

Il software che è stato realizzato permette, oltre alla cattura delle immagini da processare, di impostare i livelli di luminosità e contrasto della webcam, proprio tramite istanze della funzione `'ioctl()'`. In questo modo è possibile regolare tali parametri su valori specifici. Ad esempio, durante la fase di calibrazione vista nel capitolo 3, un contrasto non elevato permette di identificare meglio le linee che delineano il piano di lavoro, mentre, durante il normale funzionamento del sistema, un contrasto elevato sopprime le stesse, uniformandole con lo sfondo, riducendo così i disturbi.

## 5.3 La libreria Qt

Per realizzare l'interfaccia grafica del software di gestione, si è scelto di usare la libreria Qt, freeware ed open source. Questa libreria include un vasto numero di componenti (grafiche e non) e, sfruttando le proprietà della programmazione orientata agli oggetti, ciascuna di queste componenti può essere facilmente modellata attorno ai propri scopi. Nello sviluppo del software, questa libreria ha ricoperto un ruolo centrale, permettendo di gestire e visualizzare facilmente le immagini, di creare un thread dedicato all'immagine processing e di sfruttare al massimo la flessibilità offerta dalla programmazione ad eventi. Le Qt, infatti, si basano sull'architettura *signal-slot*, tramite la quale è possibile specificare quali operazioni compiere (slot) al sollevarsi di specifici eventi (signal). Questo strumento viene utilizzato in diverse occasioni all'interno del software:

- per dare inizio ai vari processamenti sulle immagini (sia in risposta alla cattura dalla webcam che alla conclusione del processamento precedente);
- per inviare al robot il comando di eseguire la ricerca della *home* o di assumere la postura desiderata;
- per variare i parametri di luminosità e contrasto della webcam;
- per dare inizio alla procedura di calibrazione.

## 5.4 La comunicazione TCP/IP

L'ultima componente del software è la comunicazione TCP/IP usata per impartire ordini al robot (client). In ambiente Unix la creazione di una connessione TCP/IP risulta molto semplice, sia lato server che client. Per quanto riguarda il lato server, il software apre un socket, cioè un flusso di dati verso la scheda di rete, che viene poi associato ad una famiglia di indirizzi IP dalle quali accettare connessioni, con il numero della relativa porta. A questo punto, dopo un'opzionale configurazione, il server rimane in attesa di connessioni da parte dei client. Non appena viene stabilita la connessione, l'indirizzo IP del client viene registrato, potendo così inviare i comandi desiderati al robot. Per quanto riguarda, invece, la connessione lato client la procedura è ancora più semplice. Il client, infatti, apre un socket che viene poi direttamente utilizzato come parametro della connessione, insieme all'indirizzo IP del server e della porta in ascolto. Non appena il server accetta la richiesta di connessione, può avvenire lo scambio di pacchetti. La lettura e la scrittura di buffer di dati sul protocollo TCP/IP avviene, come nel caso delle webcam e di semplici file, tramite istanze delle funzioni `'read()'` e `'write()'`, specificando il numero di byte che si intende scambiare.

Data la variabilità della grandezza dei dati scambiati, si è adottato un pacchetto della forma *header-data*, in cui il campo *header* di larghezza prefissata codifica la grandezza (in byte) del campo *data*.

# Capitolo 6

## Conclusioni e sviluppi futuri

Questo lavoro di tesi ha mostrato come sia possibile estrapolare dalle immagini le informazioni necessarie per far eseguire ad un robot un particolare compito. La posizione e l'orientamento di un oggetto sono informazioni necessarie in operazioni di packaging o, ancora, la forma della regione d'interesse è un'informazione indispensabile in operazioni come la saldatura. Un aspetto molto importante che questa tesi cerca di evidenziare, è l'economicità della soluzione proposta. Oggi, infatti, webcam commerciali sono disponibili a prezzi modici e la qualità offerta dalle stesse è sufficiente per realizzare il compito prefissato. Data, inoltre, la flessibilità offerta dal sistema è possibile sostituire, in qualsiasi momento, il dispositivo con uno di qualità superiore. Rimane da svolgere, comunque, molto lavoro. Sono tante, infatti, le migliorie apportabili ai vari blocchi del sistema.

Diverse tecniche di image processing possono essere implementate nell'algoritmo di calibrazione, rimuovendo le ipotesi di lavoro introdotte. Si possono integrare algoritmi di anti-distorsione, sia nella fase di calibrazione, sia durante il normale funzionamento. Esistono, inoltre, diverse tecniche di elaborazione di un'immagine a colori, potendo così recuperare i disturbi introdotti dalle diverse tipologie di lampade in commercio. Esistono tecniche di rimozione dello sfondo, che permettono di avere uno sfondo per-

fettamente bianco, semplificando di conseguenza le elaborazioni successive. Tecniche di triangolazioni permettono, a partire da due immagini provenienti da due diverse telecamere, di valutare la distanza degli oggetti, potendo così operare, ad esempio, su cumuli di oggetti, dettaglio spesso importante nelle operazioni di packaging.

L'architettura Client/Server conferisce al sistema estrema flessibilità, aprendo un ampio ventaglio di idee per sviluppi futuri. Un esempio importante è, sicuramente, la possibilità di utilizzare una sola telecamera per inquadrare una vasta area di lavoro nella quale operano diversi manipolatori, ciascuno dei quali è un client che riceve i comandi da un unico server, avente il compito di eseguire l'elaborazione delle immagini provenienti dalla telecamera stessa.

# Elenco delle figure

1.1	Imperfezioni in una saldatura evidenziate tramite sogliatura . . . . .	4
1.2	Un'immagine affetta da molto rumore ed il risultato dell'applicazione di un filtro con maschera guassiana . . . . .	6
1.3	Le principali rette in una schacchiera . . . . .	7
2.1	La fotografia di alcune monete e la rappresentazione grafica del modulo dei vettori gradiente . . . . .	10
2.2	Applicazione della sogliatura alla figura 2.1 . . . . .	10
2.3	Applicazione del filtro di Canny alla fotografia in figura 2.1 . . . . .	12
3.1	Posizione ed orientamento del piano di lavoro in un'immagine . . . . .	16
3.2	Rappresentazione vettoriale di una retta nel piano . . . . .	18
3.3	La trasformata di Hough graficamente . . . . .	20
4.1	Due manipolatori cartesiani, un antropomorfo ed uno SCARA . . . . .	23
4.2	I sistemi di riferimenti scelti . . . . .	25
4.3	Il sistema di riferimento per la risoluzione della cinematica inversa. . . . .	26
5.1	Schema a blocchi del sistema . . . . .	29

# Bibliografia

- [1] Rafael C. Gonzalez - Richard E. Woods, "*Digital image processing*" (third edition), Prentice Hall, 2008.
- [2] Bruno Siciliano - Lorenzo Sciavicco - Luigi Villani - Giuseppe Oriolo, "*Robotica - Modellistica, pianificazione e controllo*" (terza edizione), McGraw-Hill, 2008.