

Algoritmi per la navigazione di un robot autonomo ispirati agli insetti mediante visione omnidirezionale

Federico Oprandi, Alessandro Bonometti
Università degli studi di Brescia

7 febbraio 2001

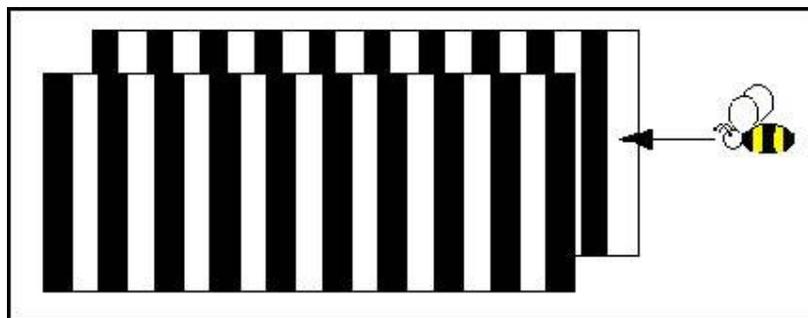
Sommario

Con questo lavoro cercheremo di descrivere un sistema originale per la navigazione di robot autonomi utilizzando il flusso ottico laterale misurato su di uno specchio conico, ispirato al criterio di stima del flusso ottico di Horn e Schunck applicato al caso monodimensionale.

1 Introduzione

Da studi entomologici sugli insetti sociali si sono scoperti meccanismi di navigazione visuale che potevano essere applicati anche nel campo della robotica avanzata [1][2]. Gli esperimenti condotti sulle api hanno messo in evidenza che, durante la navigazione attraverso un corridoio, esse tendono a volare lungo il centro di quest'ultimo. Nell'esperimento mostrato in figura 1, l'ape attraversa un corridoio condizionato da strisce bianche e nere che utilizza come riferimenti.

Sono stati condotti diversi esperimenti in tal senso, variando larghezza e frequenza delle strisce laterali; da questi si è osservato che l'ape non possiede una visione stereoscopica, bensì si basa su un sistema di visione divergente, che le permette di separare le informazioni visive laterali. Più in dettaglio, se a causa della maggiore frequenza delle strisce una parete risulta avere una maggiore velocità apparente, l'insetto se ne allontana portandosi verso quello che può essere definito il "centro ottico" del corridoio, ossia l'asse lungo cui le velocità apparenti delle pareti coincidono. Ispirandosi a questo criterio è stato creato il progetto CLF, un sistema di navigazione per robot autonomi



Volo dell'ape attraverso il corridoio

Figura 1: Schematizzazione dell'esperimento: volo dell'ape all'interno del corridoio a bande laterali

che, per simulare la visione divergente dell'ape, utilizza uno specchio conico montato coassialmente ad una telecamera [4]. Tale sistema consente, attraverso una sola telecamera, di valutare la luminosità a 360° intorno al robot, e in particolare lungo le pareti del corridoio. Attraverso opportuni calcoli, è possibile giungere ad una stima del flusso ottico laterale partendo proprio da un'immagine dello specchio. Più precisamente, sarà necessario scartare determinate zone del cono perché affette da un errore di parallasse elevato, linearizzare le immagini e mettere in opera gli opportuni algoritmi di stima del flusso, che verrà poi utilizzato per compiere le opportune scelte di navigazione. In questo elaborato sono stati utilizzati i sistemi di sempre per quanto riguarda l'acquisizione e linearizzazione delle immagini, mentre sono state introdotte nuove metodologie per quanto riguarda il calcolo del flusso ottico e la navigazione:

- Per il calcolo del flusso ottico lungo le pareti del cono è stato elaborato un algoritmo ispirato al metodo differenziale di Horn e Schunck, applicato al caso monodimensionale [5]. Tale sistema sfrutta le derivate parziali, spaziali e temporali, della luminosità lungo i settori significativi del cono in due acquisizioni consecutive.
- Per stabilizzare e irrobustire il sistema il sistema di navigazione è stato implementato un filtro di Kalman [23]. Tale algoritmo consente, data la sua natura ricorsiva e probabilistica, di ottenere buoni risultati a fronte di un limitato onere computazionale, e si rivela quindi particolarmente adatto all'elaborazione in tempo reale dei dati acquisiti.

Vedremo come l'applicazione di questo metodo sia partita dallo studio di immagini acquisite in laboratorio (limitatamente alla stima del flusso ottico), per poi approdare all'implementazione su di un robot autonomo con i relativi test di navigazione all'interno di un ambiente appositamente condizionato. Sia l'ambiente di lavoro (corridoio a bande colorate) sia gli algoritmi di navigazione sono stati fortemente influenzati dagli esperimenti sulle api, e dal comportamento di questi insetti durante i suddetti esperimenti.

2 Algoritmo basato sul metodo differenziale lineare di Horn e Schunck

2.1 La teoria del flusso ottico

Sono stati sviluppati sistemi di stima del flusso ottico basati sul gradiente (metodo differenziale), sulla correlazione di differenti regioni dell'immagine, sull'energia, etc.

Il nostro lavoro è basato su quello relativo al calcolo del gradiente associato alle variazioni di luminosità (Horn-Schunck).

Il flusso ottico è un campo vettoriale di velocità associato alle variazioni della luminosità dell'immagine [7]. Un punto fondamentale dal quale partire per sviluppare i criteri di stima del flusso ottico ispirati al gradiente è quello di considerare nulle le variazioni di luminosità di ogni singolo punto dell'immagine per variazioni infinitesimali dello spazio e del tempo; ciò equivale a parlare di *conservazione della luminosità*. Sia $\vec{p} = (x, y)$ la posizione di un generico punto P dell'immagine all'istante t e sia $\vec{u} = (u, v)$ la proiezione del vettore velocità di questo punto sul piano immagine. Dopo un tempo infinitesimo δt , l'immagine del punto P si sarà spostata alla nuova posizione $\vec{p} + \vec{u}\delta$. La conservazione della luminosità implica che [8]

$$E(\vec{p}, t) = E(\vec{p} + \vec{u}\delta t, t + \delta t)$$

dove con E si intende la luminosità del generico punto P .

Sviluppando il termine di destra della precedente equazione in serie di Taylor si ha:

$$E(\vec{p}, t) = E(\vec{p}, t) + \frac{\partial E(\vec{p}, t)}{\partial x} \frac{\partial x}{\partial t} \delta t + \frac{\partial E(\vec{p}, t)}{\partial y} \frac{\partial y}{\partial t} \delta t + \frac{\partial E(\vec{p}, t)}{\partial t} \delta t + O(\delta t^2)$$

ottenendo pertanto:

$$\frac{\partial E(\vec{p}, t)}{\partial x} \frac{\partial x}{\partial t} \delta t + \frac{\partial E(\vec{p}, t)}{\partial y} \frac{\partial y}{\partial t} \delta t + \frac{\partial E(\vec{p}, t)}{\partial t} \delta t + O(\delta t^2) = 0$$

Assumendo intervalli infinitesimi di tempo l'espressione diventa:

$$\frac{\partial E(\vec{p}, t)}{\partial x} u + \frac{\partial E(\vec{p}, t)}{\partial y} v + \frac{\partial E(\vec{p}, t)}{\partial t} = 0$$

o anche [7]

$$E_x u + E_y v + E_t = 0$$

Questa è la fondamentale equazione del vincolo associato al flusso ottico per la conservazione della luminosità. Essa implica due incognite per ogni punto del piano immagine: le componenti della velocità del flusso ottico u e v . L'espressione riscritta impiegando l'operatore gradiente diventa:

$$(\nabla E)^T \vec{u} + E_T = 0$$

Da essa emerge chiaramente che soltanto le componenti della velocità parallele al vettore gradiente della luminosità possono essere determinate dal vincolo del flusso ottico. Ne segue quindi che il campo del flusso ottico non può essere determinato unicamente con questa equazione, ma necessita di condizioni aggiuntive al fine di completare il sistema di equazioni e ricavare entrambe le componenti del flusso ottico.

2.2 Applicazione alla visione omnidirezionale

L'algoritmo sviluppato appartiene al progetto denominato CLF (*Conical Lateral Flow*), basato sul calcolo e l'elaborazione del flusso ottico dello specchio conico. Poiché ciò che interessa è il flusso laterale, fra tutti i settori del cono vengono presi in considerazione solamente quelli ritenuti utili al fine di rilevare le differenze di luminosità presenti lungo le pareti del corridoio.

Analizzando e confrontando, istante per istante, i due valori del flusso ottico laterale, il sistema di controllo deve essere in grado di stimare la posizione del robot rispetto alle pareti e, se necessario, correggerne la traiettoria.

Fra i diversi metodi esistenti per la stima del flusso ottico ci si è ispirati a quello proposto da Horn e Schunck, che si fonda sul calcolo delle derivate spaziali e temporali della luminosità.

Punto di partenza del nostro studio è l'equazione che costituisce il vincolo della conservazione dell'intensità luminosa

$$E_x u + E_y v + E_z = 0 \tag{1}$$

nella quale E_x , E_y ed E_z rappresentano le derivate parziali, spaziali e temporali, della luminosità $E(x, y, t)$, mentre u e v sono le due componenti della velocità riferite ad ogni pixel dell'immagine. Attraverso l'uso dello specchio conico si ottengono due importanti vantaggi: con una sola telecamera si ottiene un'immagine a 360° dell'ambiente circostante, e lo studio del flusso ottico, dopo un'opportuna linearizzazione, si riduce dal caso bidimensionale a quello monodimensionale. Proprio quest'ultima osservazione consente la semplificazione dell'equazione 1 nella seguente equazione differenziale lineare:

$$E_x u + E_y = 0 \tag{2}$$

Essa permette di superare la principale difficoltà nella risoluzione dell'equazione del caso bidimensionale, la ricerca cioè di vincoli aggiuntivi per determinare la soluzione di un problema sottocondizionato. Appare evidente come l'equazione 2 non necessiti di alcun vincolo aggiuntivo per giungere al valore della velocità incognita u , essendo noti E_x ed E_t .

Se da un lato l'utilizzo dello specchio conico semplifica le cose, dall'altro il suo impiego comporta una notevole perdita di informazioni oltre a un'accresciuta difficoltà nell'interpretazione dell'immagine. L'algoritmo sviluppato rappresenta quindi un tentativo di conciliare i vantaggi legati a un'equazione differenziale facilmente risolvibile con le inevitabili difficoltà causate dall'uso di uno specchio conico, con l'obiettivo di conseguire una stima del flusso ottico coerente con le traiettorie acquisite in laboratorio.

2.3 Implementazione dell'algoritmo

2.3.1 Coefficienti per l'errore di parallasse

Nell'utilizzo dello specchio conico è necessario considerare l'errore di parallasse: immaginando di suddividere lo specchio in 360 settori uguali, si intuisce come ogni settore raccolga la luce proveniente dalle pareti circostanti in relazione alla distanza e all'angolazione rispetto ad essa. In riferimento alla Figura 2, è possibile comprendere come la porzione di parete "catturata" dal settore 1 sia inferiore rispetto a quella catturata dal settore 2, e proprio per questa ragione si sono calcolati i coefficienti dell'errore di parallasse. Con semplici formule trigonometriche, e osservando la Figura 3, si ricava la lunghezza del generico tratto di parete associato all' n -esimo settore dello specchio conico, in funzione della distanza d del centro del cono dalla parete:

$$\begin{aligned}\frac{S_0}{2} &= d \tan(0,5^\circ) \Rightarrow S_0 = 2d \tan(0,5^\circ) \\ S_1^* &= d \tan(1^\circ + 0,5^\circ) = d \tan(1,5^\circ) \\ S_1 &= S_1^* - \frac{S_0}{2} = d [\tan(1,5^\circ) - \tan(0,5^\circ)]\end{aligned}$$

Generalizzando:

$$S_n = d \{ \tan [(n,5)^\circ] - \tan [(n-1,5)^\circ] \}$$

Per eliminare il valore d dall'espressione degli S_n si sono divisi tutti i coefficienti per il loro valore minimo, ossia il valore S_0 . E' stato definito con pedice 0 il coefficiente relativo al settore dello specchio la cui mediana è perpendicolare alla parete, con pedice 1 quello relativo al settore appartenente

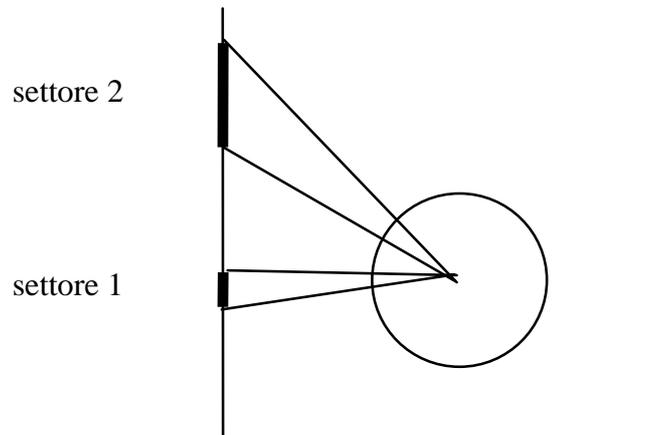


Figura 2: A seconda della posizione, angoli uguali “spazzano” lunghezze differenti di corridoio

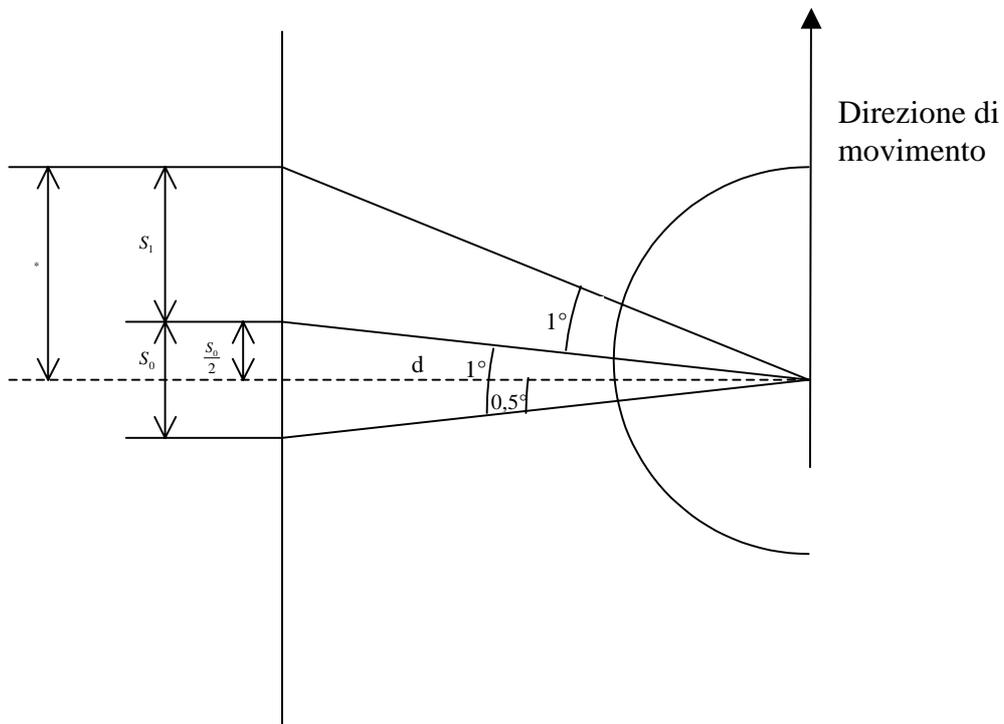


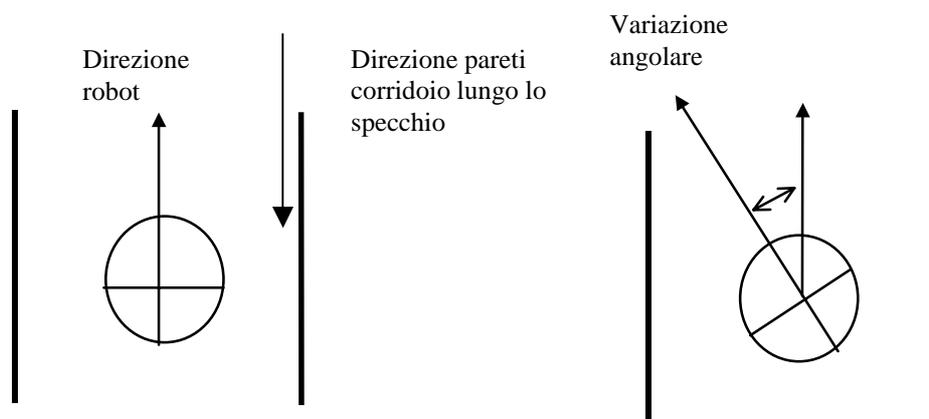
Figura 3: Calcolo dell'errore di parallasse

alla parte superiore del cono (nel verso cioè della direzione di movimento), la cui mediana forma un angolo di 1° con la normale alla parete, e con pedice -1 quello appartenente alla parte inferiore del cono (nel verso opposto alla direzione di movimento) la cui mediana forma un angolo di un grado con la normale alla parete.

Procedendo in questo modo si ricavano, oltre al coefficiente S_0 , 89 coefficienti (S_1, S_2, \dots, S_{89}) associati agli 89 settori della parte superiore del cono, e 89 coefficienti ($S_{-1}, S_{-2}, \dots, S_{-89}$) associati ai rispettivi settori della metà inferiore del cono. Appare ovvio, dalla simmetria della figura, che $S_x = S_{-x}$, con $1 \leq x \leq 89$, oltre al fatto che il calcolo di tali coefficienti è equivalente per entrambi i lati dello specchio conico.

2.3.2 Come opera il metodo di stima del flusso ottico

Punto fondamentale che caratterizza questo algoritmo è il supporre che vi sia un sistema in grado di riconoscere le variazioni angolari della direzione di movimento del robot rispetto alla direzione in linea con le pareti del corridoio, come dalla seguente figura:



Questo vincolo, decisamente restrittivo, è però di grande importanza in quanto l'algoritmo assegna un peso alle variazioni di intensità luminosa di ogni settore del cono a seconda dell'errore di parallasse.

Ad esempio il settore 0 dello specchio conico di Figura 4, nel caso fosse avvenuta una rotazione antioraria di 2 gradi, catturerà la luce di una porzione di parete equivalente a quella che in precedenza veniva catturata dal settore 2; tale variazione viene espressa dall'algoritmo attribuendo un peso differente ad ogni settore secondo un assegnamento dinamico che tenga conto, istante per istante, dello sfasamento angolare fra la direzione del robot e quella delle pareti del corridoio.

Per la stima del flusso ottico ci siamo serviti di 132 settori per ogni lato del cono, distribuiti come illustrato nella Figura 4. La ragione per cui si sono considerati 66 settori nella parte inferiore del cono e 65 in quella superiore dipende dall'espressione della derivata spaziale discreta implementata dal metodo di stima del flusso e verrà meglio chiarita in seguito.

Ciò che abbiamo voluto ricavare dall'algoritmo è un vettore contenente i valori del flusso ottico di ogni settore del lato destro, e un vettore per i valori del flusso ottico dei settori del lato sinistro.

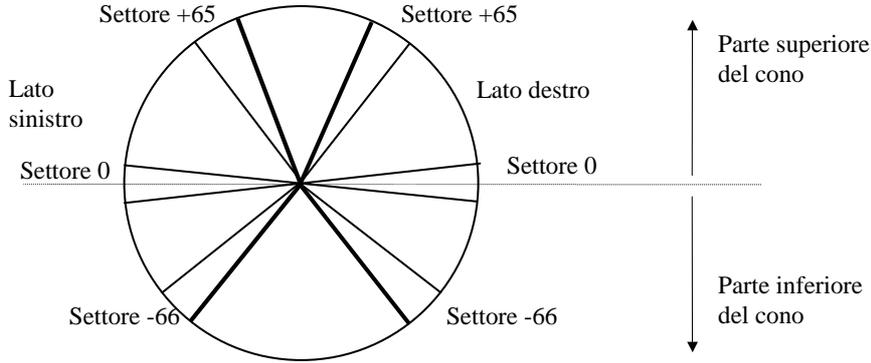


Figura 4: La numerazione dei settori del cono

Ricordando l'espressione della stima monodimensionale del flusso ottico (Equazione 2):

$$E_x u + E_t = 0$$

l'espressione dell'incognita u diventa:

$$u = -\frac{E_t}{E_x} = -\frac{\partial E_t}{\partial t} \frac{\partial x}{\partial E_x} \quad (3)$$

o, riscritta in una simbologia più adatta alla comprensione del nostro problema:

$$u_n(t) = -\frac{\partial E_{tn}(t)}{\partial t} \frac{\partial x_n}{\partial E_{xn}(t)} \quad (4)$$

Laddove si consideri che la direzione di movimento del robot è nota, è possibile considerare $u_n(t)$ in valore assoluto, ottenendo quindi la seguente espressione:

$$u_n(t) = \left| -\frac{\partial E_{tn}(t)}{\partial t} \frac{\partial x_n}{\partial E_{xn}(t)} \right| \quad (5)$$

Per ricavare la stima del flusso ottico un del generico settore n dello specchio conico, siamo dovuti ricorrere alla definizione di particolari derivate

discrete, spaziali e temporali, dell'intensità luminosa $E(x, t)$. Poiché l'algoritmo opera in maniera indipendente sui settori di destra e su quelli di sinistra, l'espressione è da intendersi valida per entrambi i lati del cono. Abbiamo pertanto definito in questo modo le variabili dell'equazione 4:

- $u_n(t)$: Flusso ottico associato al settore n all'istante t .
- $\frac{\partial E_{t_n}(t)}{\partial t}$: Derivata discreta rispetto al tempo, calcolata all'istante t , della luminosità del settore n . Per definire tale derivata abbiamo assegnato per comodità valore unitario al tempo di campionamento delle immagini, $dt = 1$, mentre per esprimere dE_{t_n} abbiamo posto semplicemente:

$$\partial E_{t_n}(t) = E_n(t) - E_n(t - 1)$$

ossia ∂E_{t_n} è la differenza di luminosità del settore n rilevata in due istanti consecutivi.

- $\frac{\partial E_{x_n}(t)}{\partial x_n}$: Derivata discreta rispetto allo spazio, calcolata all'istante t , della luminosità del settore n . La derivata spaziale in una dimensione deve tenere conto della deformazione operata dallo specchio conico, ed è quindi necessario considerare, per ciascun settore, l'angolazione rispetto alle pareti del corridoio. Ciò è realizzato attraverso i coefficienti dell'errore di parallasse: per meglio comprendere il significato di questa derivata la Figura 5 mostra la luminosità dei vari settori, la cui ampiezza (asse x) dipende dall'errore di parallasse.

$\partial E_{x_n}(t)$ corrisponde alla differenza di luminosità, all'istante t , fra il settore n e il settore $n - 1$:

$$\partial E_{x_n}(t) = E_n(t) - E_{n-1}(t)$$

∂x_n è il generico spostamento differenziale ∂x associato ad ogni derivata e varia a seconda del settore considerato (Figura 5). Si noti che l'ampiezza di ogni settore S_n sull'asse delle ascisse corrisponde esattamente ai coefficienti dell'errore di parallasse C_n elencati nell'Appendice A a pagina 53.

Ad esempio, la derivata spaziale corrispondente al settore 1 sarà

$$\frac{E_1(t) - E_0(t)}{C_1/2 + C_0/2}$$

con

$$C_1/2 + C_0/2 = \partial x_1$$

mentre la derivata corrispondente al settore 65 sarà

$$\frac{E_{65}(t) - E_{64}(t)}{C_{65}/2 + C_{64}/2}$$

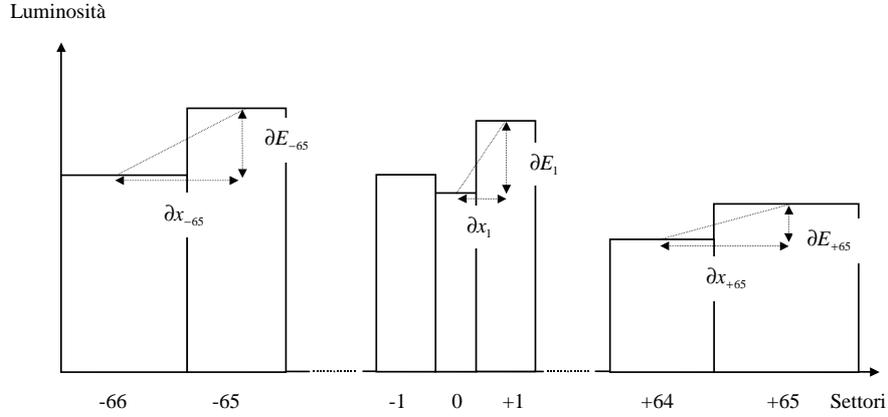


Figura 5: “espansione” o “compressione” dei settori in corrispondenza dell’errore di parallasse

con

$$C_{65}/2 + C_{64}/2 = \partial x_{65}$$

così come quella del settore -65 sarà

$$\frac{E_{-65}(t) - E_{-66}(t)}{C_{-65}/2 + C_{-66}/2}$$

con

$$C_{-65}/2 + C_{-66}/2 = \partial x_{-65}$$

Queste due ultime espressioni suggeriscono il motivo per cui si sono considerati 66 settori nella parte inferiore del cono e 65 in quella superiore: la derivata è infatti data dalla differenza dell’intensità luminosa di un settore e del precedente, e se il precedente del settore -65 è il -66, il precedente del settore +65 è il +64; da qui la necessità di introdurre un settore (il -66) che serve unicamente per il calcolo della derivata spaziale del settore successivo.

Nel caso avvenga una variazione angolare di direzione rispetto a quella parallela con le pareti del corridoio, i pesi associati ai settori muteranno e di conseguenza cambieranno le espressioni dei vari ∂x_n .

Come esempio si supponga che il robot compia una rotazione verso destra di 2 gradi; il settore 0 di destra, rappresentato in Figura 4 a pagina 9, solidalmente con il cono ruoterà di 2 gradi raccogliendo una porzione di intensità luminosa precedentemente catturata dal settore -2. L’espressione del flusso ottico attribuito al settore 0 rimarrà invariata in ogni sua parte, eccezion fatta per il ∂x_0 ; questo passerà da $C_0/2 + C_{-1}/2$ a $C_{-2}/2 + C_{-3}/2$. Tutti i ∂x del lato destro subiranno una sorta di traslazione all’indietro di 2, motivo per il

quale si avrà il passaggio di ∂x_{-65} da $C_{-66}/2 + C_{-65}/2$ a $C_{-68} + C_{-67}$, mentre per quando riguarda ∂x_{65} , esso passerà da $C_{65}/2 + C_{64}/2$ a $C_{63}/2 + C_{62}/2$.

Data la distribuzione crescente, ma non regolare, dei coefficienti di parallasse (i coefficienti da C_0 a C_{-76}) assumono valori compresi fra 1 e 17 mentre i coefficienti restanti da C_{-77} a C_{-89} assumono valori da 19 a 4377) al fine di evitare errori dovuti a settori con errore di parallasse troppo elevato, si è pensato di considerare solo i coefficienti fino al assumono valori da 19 a 4377) al fine di evitare errori dovuti a settori con errore di parallasse troppo elevato, si è pensato di considerare solo i coefficienti fino al C_{-76} . Così facendo è possibile calcolare fino a $dx_{-75} = C_{-76}/2 + C_{-75}/2$ ma non oltre; ciò significa che, per rotazioni fino a 10 gradi, si è in grado di assegnare una stima del flusso ottico ad ogni settore.

Tuttavia, in caso di una rotazione superiore ai 10 gradi, ad esempio una rotazione verso sinistra di 11 gradi, al settore 64 viene assegnato un $dx_{64} = C_{76}/2 + C_{75}/2$ mentre per il settore 65 non si esegue il calcolo del flusso ottico (poiché manca il coefficiente C_{77}) e di conseguenza il flusso ottico del settore 65 non contribuirà alla stima del flusso ottico destro totale. Chiaramente, qualora la rotazione superiore ai 10 gradi fosse verso destra, i settori del lato destro che verrebbero ignorati nella stima del flusso ottico totale sarebbero quelli della metà inferiore dello specchio conico. Immaginando una rotazione verso destra di 14 gradi, questa comporterebbe l'eliminazione dei settori -65, -64, -63 e -62.

Quanto è stato detto per il lato destro dello specchio conico è ovviamente valido anche per il lato sinistro, con l'unica avvertenza di considerare che una qualsiasi rotazione produce effetti esattamente opposti sui due lati : riferendosi all'ultimo esempio, i 4 settori di sinistra ad essere scartati sarebbero i 62, 63, 64 e 65. La Figura 6 evidenzia i settori eliminati.

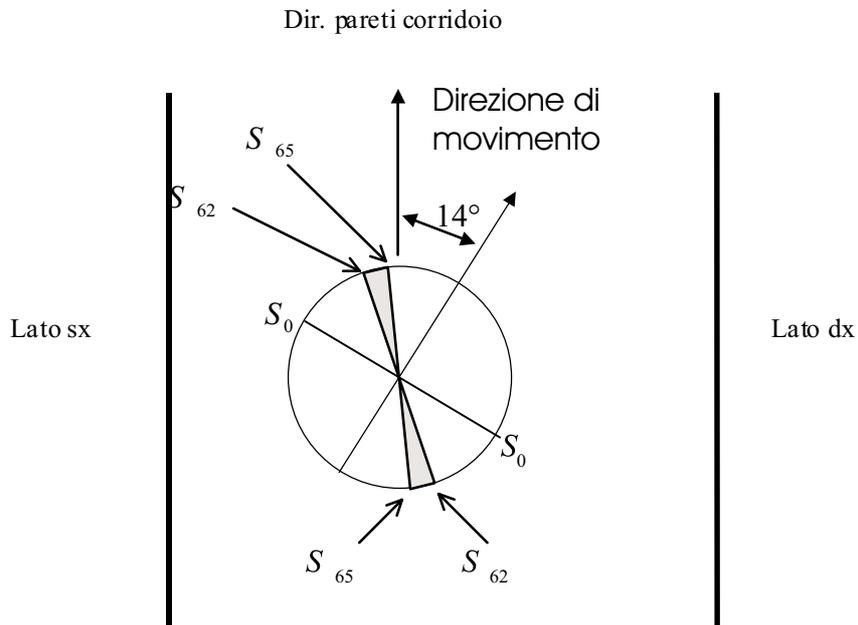


Figura 6: I settori del cono trascurati in seguito alla rotazione del robot

3 Test

Al fine di verificare la validità di entrambi i metodi ci siamo avvalsi di 50 immagini acquisite nel laboratorio di robotica avanzata all'interno di un corridoio di 5 metri per 2 creato utilizzando delle pareti nere con bande colorate equidistanti (Figura 7). Le immagini sono state prelevate lungo 4 traiettorie molto precise e ben distinte fra loro con lo scopo di ritrovare tali differenze anche nelle informazioni associate al flusso ottico:

Le acquisizioni sono state effettuate in questo modo:

- 13 acquisizioni a 50 cm dal lato destro e a 150 cm dal lato sinistro,



Figura 7: Schematizzazione di una parete del corridoio

ognuna distanziata di 25 cm.

- 13 acquisizioni a 150 cm dal lato destro e a 50 cm dal lato sinistro, ognuna distanziata di 25 cm.
- 13 acquisizioni a 1 metro dal lato destro e a 1 metro dal lato sinistro, ognuna distanziata di 25 cm.
- 11 acquisizioni sulla diagonale congiungente un punto a 50 cm dal lato destro e un punto a 50 cm dal lato sinistro dopo un avanzamento di 3 metri rispetto alla direzione delle pareti del corridoio .

Le traiettorie seguite sono schematizzate in Figura 8.

Entrambi i sistemi di navigazione si propongono di consentire la navigazione di un robot al centro di un corridoio rettilineo, esaminando le immagini acquisite da una telecamera posta sopra uno specchio conico (Figura 9).

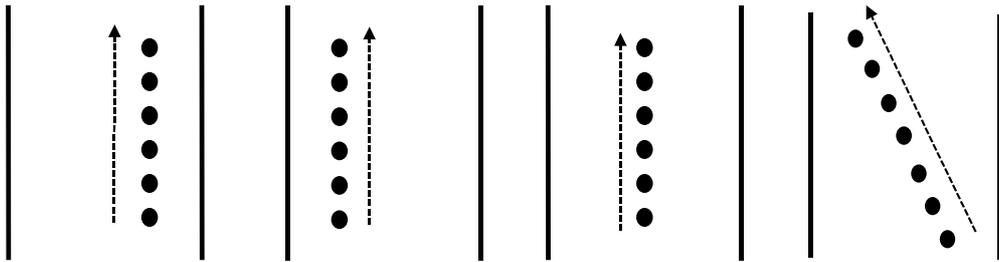


Figura 8: Le 4 traiettorie lungo cui sono state prelevate le immagini (rispettivamente destro, sinistro, centrale, diagonale)

L'immagine ottenuta dallo specchio viene suddivisa in 360 settori identici e successivamente linearizzata (Figura 10).

3.1 Analisi delle acquisizioni

L'analisi dei dati raccolti in laboratorio è stata effettuata con il programma Matlab. Il nucleo dell'algoritmo è contenuto nel file `flusso.m`, il quale richiede in ingresso due files contenenti i valori RGB di due immagini consecutive. Più precisamente, ogni file contiene un vettore colonna, denominato M , di 1080 elementi; nelle prime 360 posizioni vi sono i valori del rosso, quindi i valori del verde e le ultime 360 per i valori del blu. L'ordine di riempimento del vettore è quello indicato in Figura 11, e dalla linearizzazione si ottiene il vettore di Figura 12.

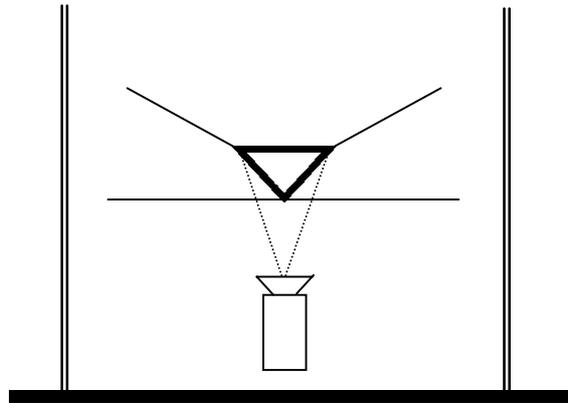


Figura 9: La posizione reciproca di cono e telecamera

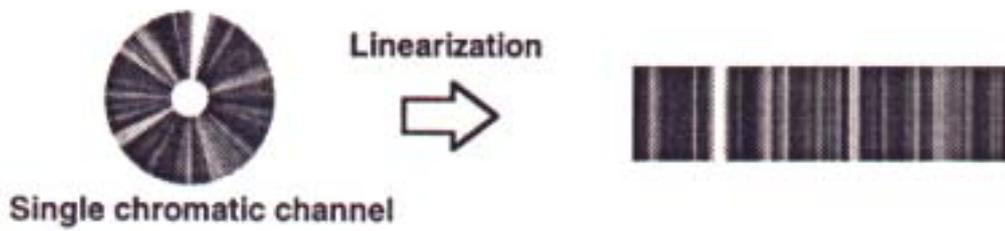


Figura 10: Lo schema di linearizzazione

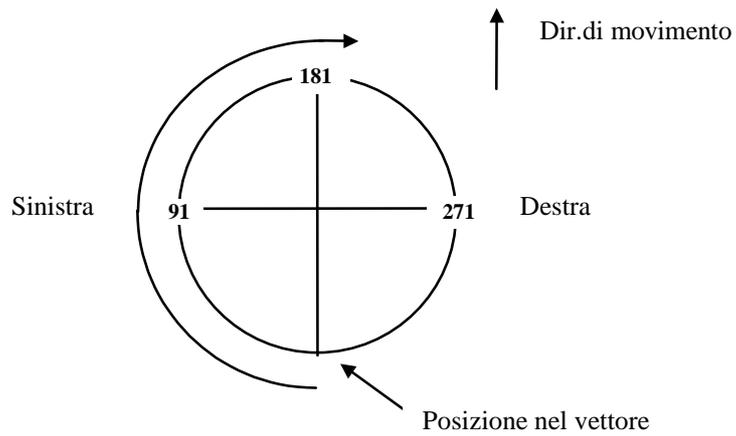


Figura 11: La direzione di riempimento del vettore

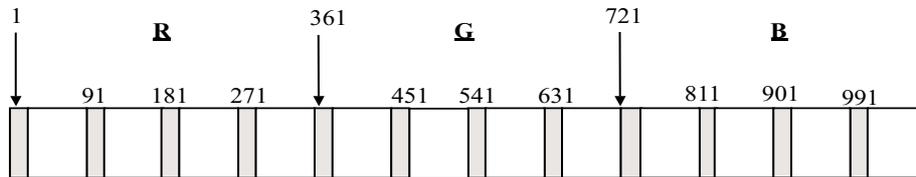
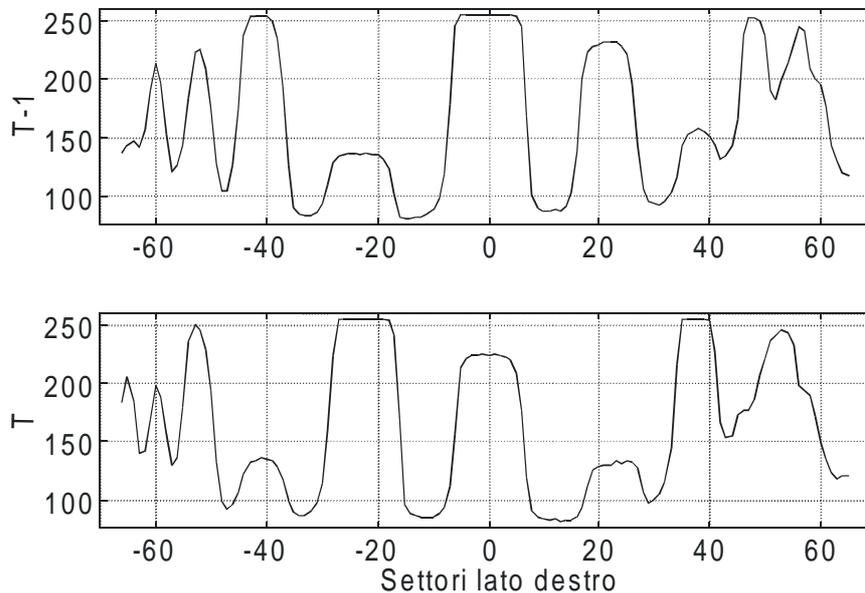


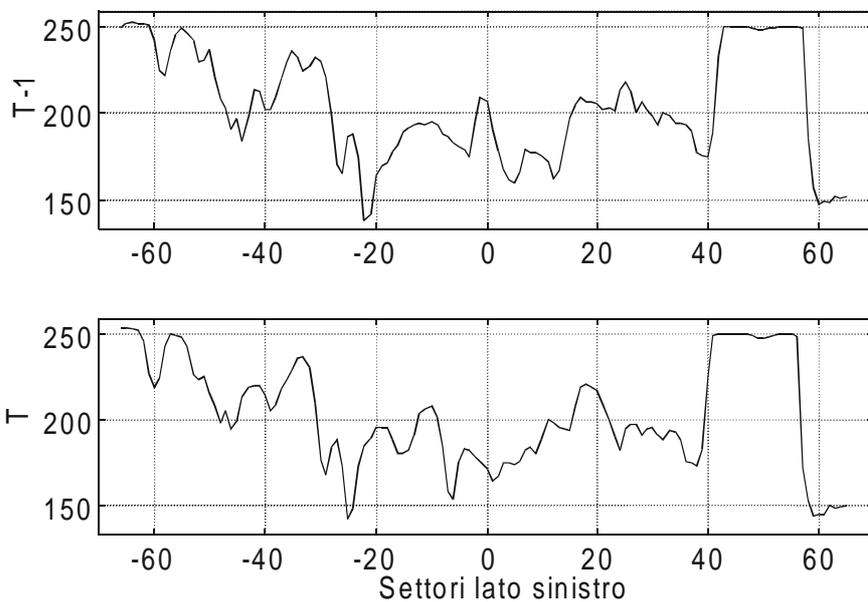
Figura 12: Corrispondenza tra gli indici del vettore e la posizione dei settori sullo specchio conico

Inoltre l'algoritmo prende in considerazione l'angolo di deviazione rispetto all'asse del corridoio, consentendo così di elaborare traiettorie diagonali.

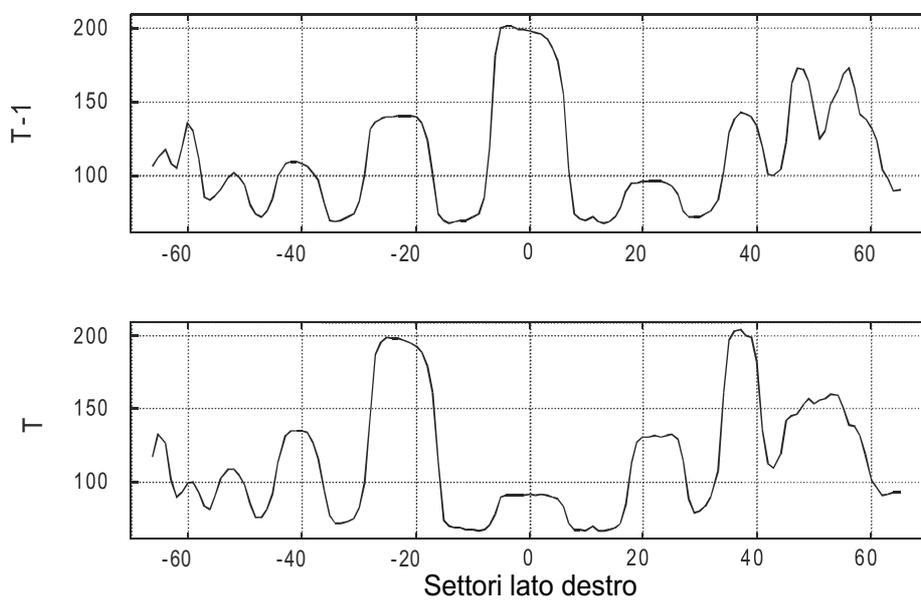
I grafici seguenti mostrano un esempio del flusso ottico destro e sinistro; i dati visualizzati sono il risultato della stima effettuata su due immagini acquisite in vicinanza della parete destra. Inizialmente vengono illustrati i valori della luminosità acquisiti per ogni componente della terna RGB, sia a destra sia a sinistra, seguiti successivamente dalla rappresentazione grafica dei rispettivi valori del flusso ottico..



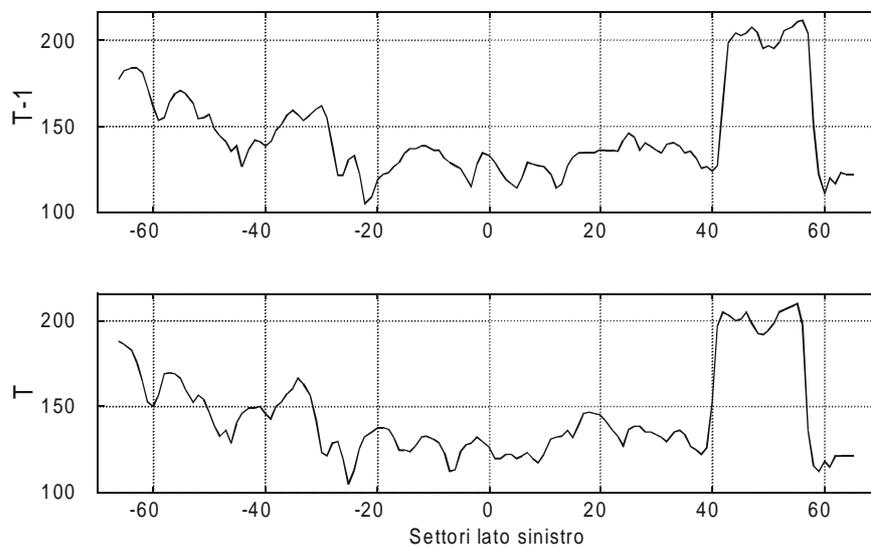
L'intensità luminosa del rosso lungo i settori significativi di destra del cono, in due acquisizioni consecutive



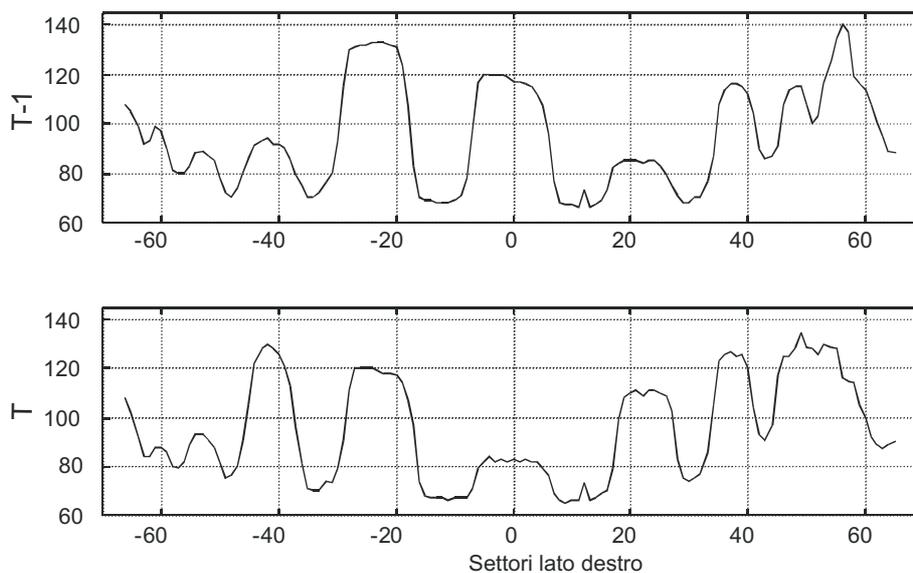
L'intensità luminosa del rosso lungo i settori significativi di sinistra del cono, in due acquisizioni consecutive



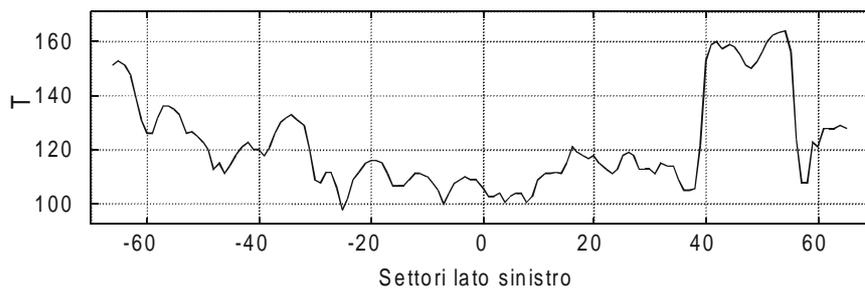
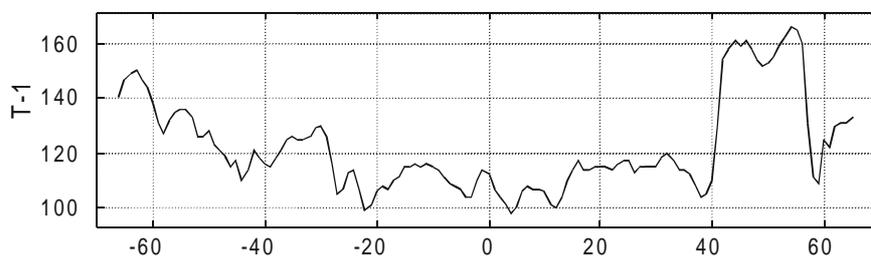
L'intensità luminosa del verde lungo i settori significativi di destra del cono, in due acquisizioni consecutive



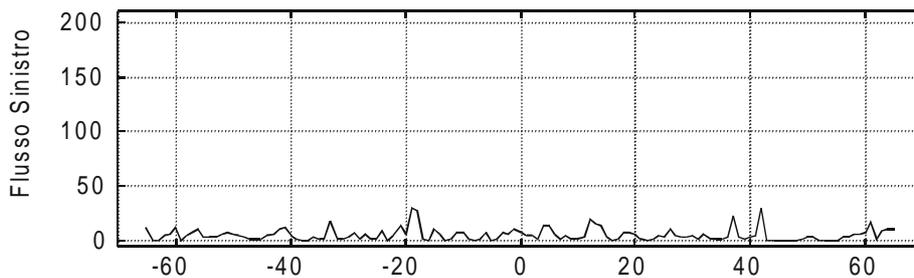
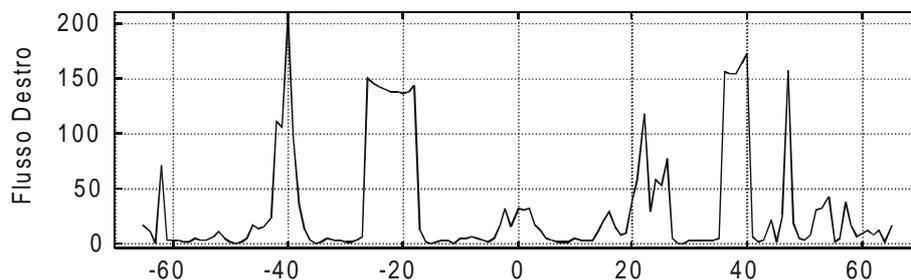
L'intensità luminosa del verde lungo i settori significativi di sinistra del cono, in due acquisizioni consecutive



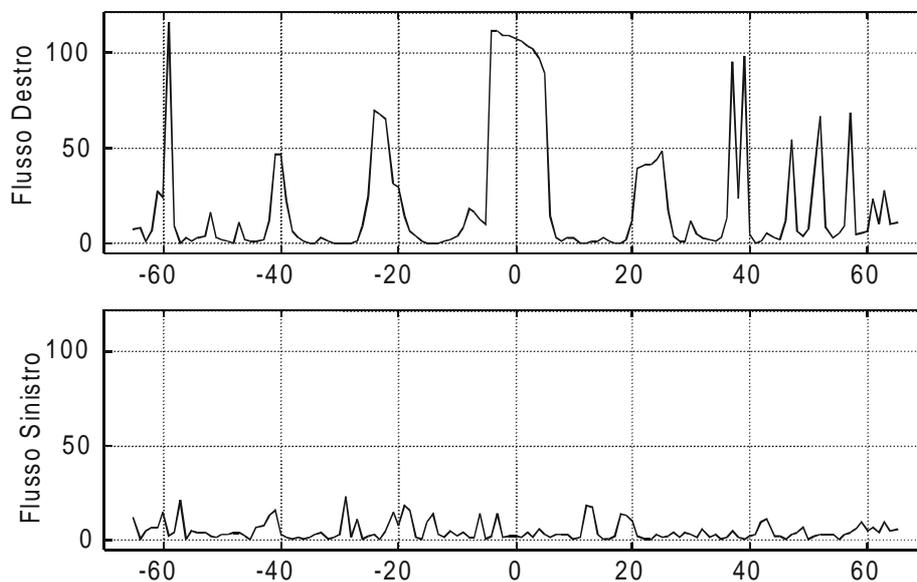
L'intensità luminosa del blu lungo i settori significativi di destra del cono, in due acquisizioni consecutive



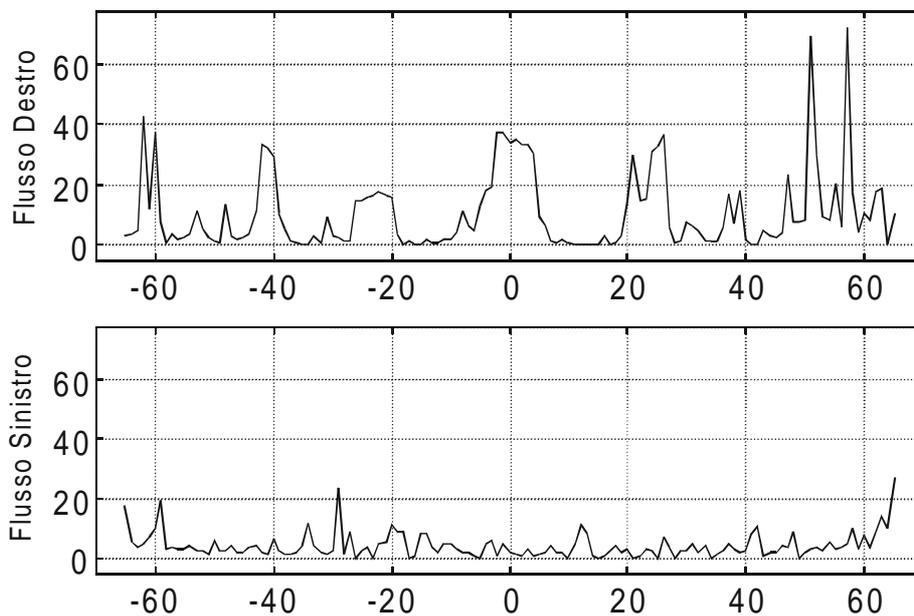
L'intensità luminosa del blu lungo i settori significativi di sinistra del cono, in due istanti consecutivi



Il flusso ottico del rosso, settore per settore, lungo i lati destro e sinistro del cono



Il flusso ottico del verde, settore per settore, lungo i lati destro e sinistro del cono



Il flusso ottico del blu, settore per settore, lungo i lati destro e sinistro del cono

Una volta ricavati i sei vettori contenenti le stime del flusso ottico relative ad ogni settore, l'algoritmo calcola il valore del flusso ottico complessivo

destro e sinistro, operando nel seguente modo: al generico settore n , supponiamo appartenente al lato destro, si assegna la stima del flusso ottico complessivo calcolata come media geometrica dei tre valori del flusso precedentemente ottenuti per questo settore del lato destro nelle componenti RGB.

Si giunge pertanto a due vettori, uno per ciascun lato dello specchio, ognuno dei quali contiene le 131 stime complessive (65 settori positivi + 65 settori negativi + 1 settore 0) del flusso ottico relative ad ogni settore; infine si perviene ai due valori del flusso ottico complessivo destro e sinistro attraverso la media dei valori appartenenti ai due vettori. Per chiarire maggiormente quanto appena esposto, ci avvarremo del formalismo matematico. Definendo:

- $RFdx_n$ il Flusso F del settore n destro (dx) relativo al colore Rosso (R)
- $GFdx_n$ il Flusso F del settore n destro (dx) relativo al colore Verde (G)
- $BFdx_n$ il Flusso F del settore n destro (dx) relativo al colore Blu (B)

si ottiene il flusso complessivo del settore destro come

$$Fdx_n = \sqrt[3]{RFdx_n * GFdx_n * BFdx_n}$$

e dualmente il flusso sinistro come

$$Fsx_n = \sqrt[3]{RFsx_n * GFsx_n * BFsx_n}$$

con definizioni analoghe a quelle dell'altro lato.

Ecco quindi che i flussi complessivi, destro e sinistro, si ottengono rispettivamente come:

$$Fdx = \frac{Fdx_{-65} + Fdx_{-64} + \dots + Fdx_0 + \dots + Fdx_{64} + Fdx_{65}}{131}$$

$$Fsx = \frac{Fsx_{-65} + Fsx_{-64} + \dots + Fsx_0 + \dots + Fsx_{64} + Fsx_{65}}{131}$$

Poiché l'obbiettivo ultimo dell'algoritmo è quello di rilevare la differenza fra il valore destro e sinistro del flusso al fine di guidare un robot correggendo la traiettoria, il processo di stima si conclude calcolando i valori normalizzati di Fdx e Fsx rispetto al maggiore fra i due, ossia se $Fdx > Fsx$ allora

$$Fdx = 1, Fsx = \frac{Fsx}{Fdx}$$

Viceversa se $Fsx > Fdx$ allora

$$Fsx = 1, Fdx = \frac{Fdx}{Fsx}$$

3.2 Risultati sperimentali

Per testare la validità del metodo abbiamo eseguito 4 prove, osservando quindi i risultati ricavati per le 4 traiettorie mostrate in Figura 8. I dati ottenuti dalle acquisizioni sono stati inseriti in appositi file la cui organizzazione è stata descritta a partire da pagina 14. Per una corretta lettura e interpretazione dei risultati è necessario considerare almeno questi tre fattori che hanno condizionato l'idealità dell'acquisizione:

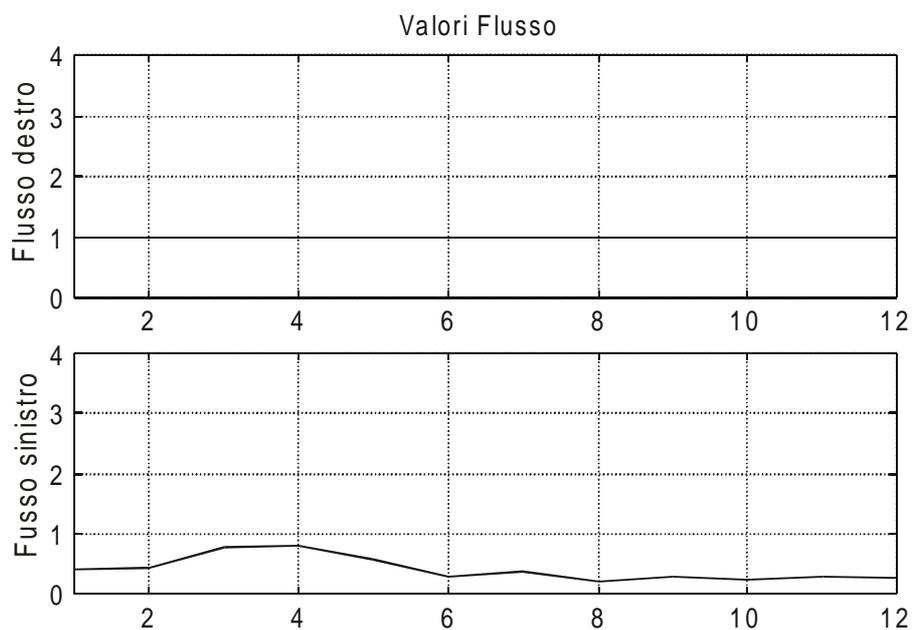
- Errori di piazzamento dello specchio conico: imprecisioni nel rispettare l'equidistanza fra un'acquisizione e difficoltà nel mantenere lo specchio conico con un orientamento costante rispetto alle pareti del corridoio (difficoltà riscontrata soprattutto nel corso della quarta prova).
- Pareti del corridoio non perfettamente rettilinee : essendo di cartone presentavano in alcune parti delle deformazioni.
- Illuminazione non uniforme : data la presenza contemporanea di finestre (parzialmente nascoste da tende oscuranti) e di lampade al neon (non "centrate" rispetto al corridoio), taluni punti delle pareti erano caratterizzati da zone d'ombra.

3.2.1 1.a prova: traiettoria n°1

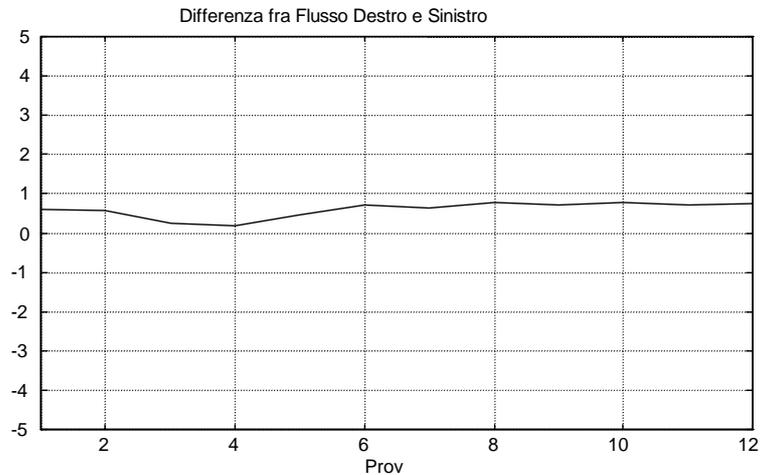
La tabella seguente illustra i risultati ottenuti al termine della prova; poiché si è sempre mantenuto lo specchio conico parallelo alla direzione del corridoio, l'angolo di sfasamento (o la variazione angolare, vedi Figura 6 a pagina 13) in questo esperimento è nullo.

Angolo di sfasamento	Stima Flusso destro (F_{dx})	Stima Flusso sinistro (F_{sx})	Differenza ($F_{dx} - F_{sx}$)
0	1	0.3848	0.6152
0	1	0.4215	0.5785
0	1	0.7619	0.2381
0	1	0.8040	0.1960
0	1	0.5561	0.4439
0	1	0.2792	0.7208
0	1	0.3663	0.6337
0	1	0.2118	0.7882
0	1	0.2745	0.7255
0	1	0.2236	0.7764
0	1	0.2934	0.7066
0	1	0.2443	0.7557

Tabella 1: I valori del flusso destro e sinistro, e la loro differenza, relativi ad ogni passo lungo la traiettoria di destra



Il flusso complessivo nell'arco delle dodici prove sui lati destro e sinistro



La differenza tra flusso destro e sinistro nell'arco delle dodici prove lungo il lato destro del corridoio

Osservando la tabella e i grafici sopra emerge immediatamente una netta prevalenza, in tutti e dodici gli esperimenti, del flusso laterale destro; questo è in accordo con quanto ci si attendeva dalla teoria.

Purtroppo l'andamento delle due curve non è regolare, a causa di una stima non costante dei due flussi ottici, dovuta probabilmente ai succitati errori che in fase sperimentale hanno condizionato le acquisizioni.

Inoltre è necessario considerare che l'algoritmo si fonda sul vincolo della conservazione della luminosità; ciò implica che per applicare nel modo più corretto possibile questo metodo di stima del flusso ottico è necessario che le variazioni di luminosità fra due acquisizioni consecutive siano infinitesime, in altre parole lo specchio conico avrebbe dovuto acquisire immagini per spostamenti molto piccoli.

Evidentemente i 25 centimetri fra un campionamento e l'altro si sono rivelati non sufficienti per consentire all'algoritmo di operare nelle migliori condizioni, e le notevoli variazioni delle immagini in due istanti successivi si possono bene osservare a partire da pagina 16.

Per le stesse ragioni, tale andamento irregolare si è verificato anche nelle rimanenti tre prove; non volendo essere ripetitivi, in seguito non sottolineeremo più la mancanza di linearità nell'andamento della stima dei flussi ottici laterali, mentre si valuteranno i risultati unicamente in base al confronto fra il flusso destro e quello sinistro.

3.2.2 2.a prova: traiettoria n°2

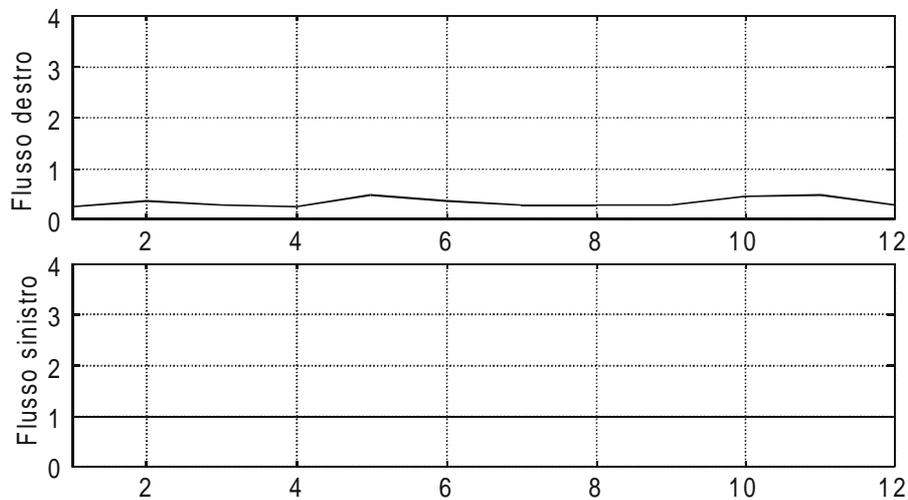
La tabella seguente illustra i risultati ottenuti al termine della prova; poiché si è sempre mantenuto lo specchio conico parallelo alla direzione del corridoio,

Angolo di sfasamento	Stima Flusso destro (F_{dx})	Stima Flusso sinistro (F_{sx})	Differenza ($F_{dx} - F_{sx}$)
0	0.2504	1	-0.7496
0	0.3613	1	-0.6387
0	0.2826	1	-0.7174
0	0.2490	1	-0.7510
0	0.4710	1	-0.5290
0	0.3797	1	-0.6203
0	0.2874	1	-0.7126
0	0.2781	1	-0.7219
0	0.2797	1	-0.7203
0	0.4545	1	-0.5455
0	0.4881	1	-0.5119
0	0.2769	1	-0.7231

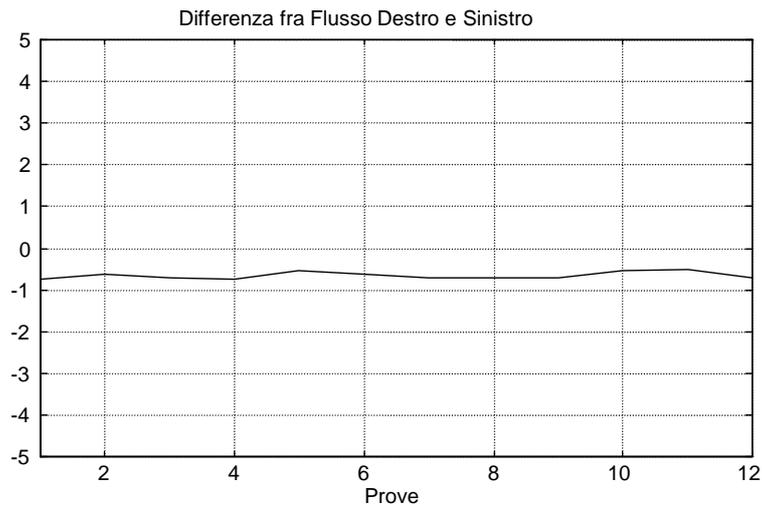
Tabella 2: I valori del flusso destro e sinistro, e la loro differenza, relativi ad ogni passo lungo la traiettoria di sinistra

l'angolo di sfasamento (o la variazione angolare, vedi Figura 6 a pagina 13) in questo esperimento è nullo

Il grafici seguenti mostrano il confronto fra la stima del flusso ottico laterale destro e la stima del flusso sinistro nell'arco dei 12 esperimenti. Così come è avvenuto nella prova precedente, con la seconda traiettoria si è verificata la prevalenza dei valori di un flusso sui valori dell'altro flusso; essendo risultati derivanti da immagini prelevate in prossimità della parete sinistra, coerentemente con quanto ci si attendeva, le stime del flusso laterale sinistro sono maggiori di quelle di destra in tutti e dodici gli esperimenti.



Il flusso complessivo nell'arco delle dodici prove sui lati destro e sinistro



La differenza tra flusso destro e sinistro nell'arco delle dodici prove lungo il lato sinistro del corridoio

3.2.3 3.a prova: traiettoria n°3

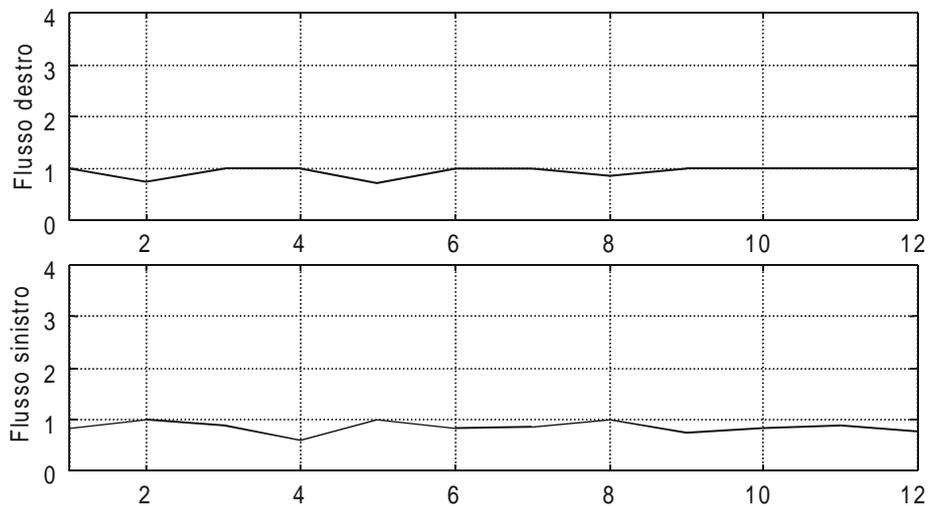
La tabella seguente illustra i risultati ottenuti al termine della prova; poiché si è sempre mantenuto lo specchio conico parallelo alla direzione del corridoio, l'angolo di sfasamento (o la variazione angolare, vedi Figura 6 a pagina 13) anche in questo esperimento è nullo.

I grafici che seguono illustrano il confronto fra la stima del flusso ottico laterale destro e la stima del flusso sinistro nell'arco dei 12 esperimenti. L'a-

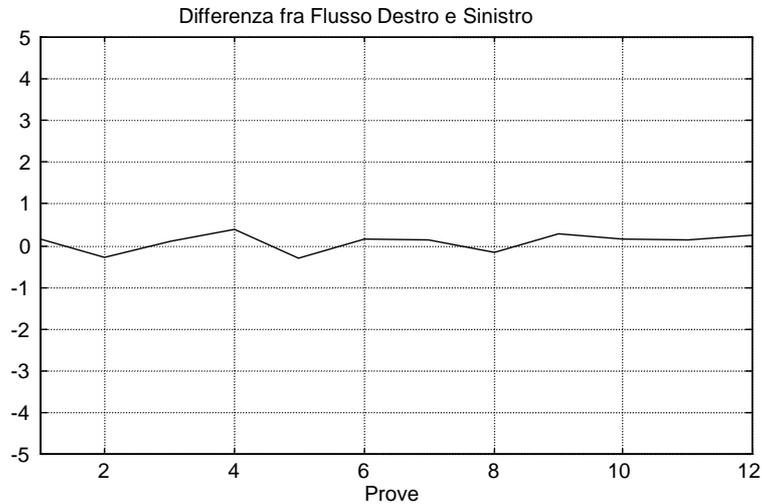
Angolo di sfasamento	Stima Flusso destro (F_{dx})	Stima Flusso sinistro (F_{sx})	Differenza ($F_{dx} - F_{sx}$)
0	1	0.8316	0.1684
0	0.7273	1	-0.2727
0	1	0.8921	0.1079
0	1	0.6021	0.3979
0	0.6935	1	-0.3065
0	1	0.8278	0.1722
0	1	0.8546	0.1454
0	0.8452	1	-0.1548
0	1	0.7316	0.2684
0	1	0.8271	0.1729
0	1	0.8701	0.1299
0	1	0.7580	0.2420

Tabella 3: I valori del flusso destro e sinistro, e la loro differenza, relativi ad ogni passo lungo la traiettoria centrale

nalisi del primo grafico evidenzia la somiglianza fra i valori delle due curve di stima del flusso ottico destro e sinistro, ed è utile osservare come la differenza (secondo grafico) tra le due curve oscilli approssimativamente attorno al valore 0.



Il flusso complessivo nell'arco delle dodici prove sui lati destro e sinistro



La differenza tra flusso destro e sinistro nell'arco delle dodici prove lungo il centro del corridoio

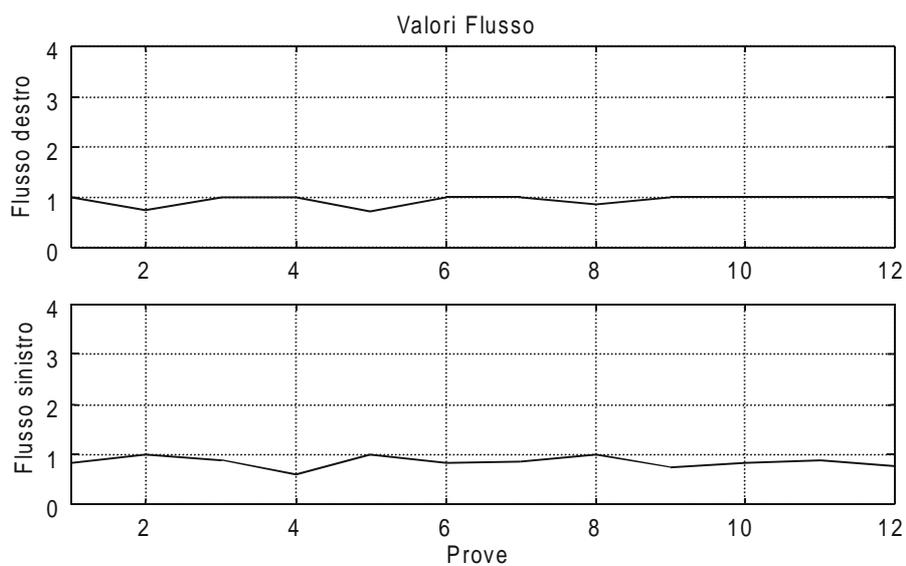
3.2.4 4.a prova: traiettoria n°4

La tabella seguente illustra i risultati ottenuti al termine della prova; poiché si è sempre mantenuto lo specchio conico orientato lungo una traiettoria formante un angolo di circa 18° , cioè $\arctan\left(\frac{1}{3}\right)$, con la direzione del corridoio, l'angolo di sfasamento in questa prova è stato posto uguale a 18° . Essendo l'algoritmo di natura discreta, i decimali di grado vengono sistematicamente ignorati; per rotazioni antiorarie i valori dell'angolo di sfasamento sono arrotondati per difetto (18,43 diventa 18), mentre per rotazioni orarie i valori dell'angolo di sfasamento sono arrotondati per eccesso (-18,43 diventa -18).

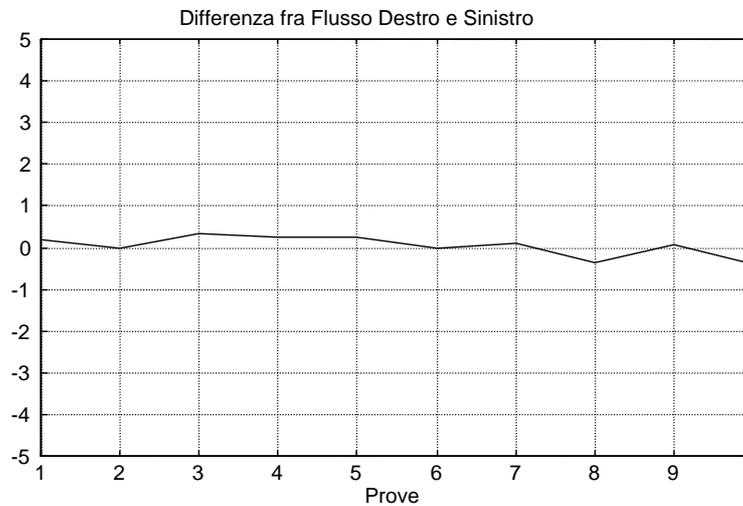
Il grafico sottostante mostra il confronto fra la stima del flusso ottico laterale destro e la stima del flusso sinistro nell'arco dei 10 esperimenti, seguito quindi dall'andamento della differenza fra le due stime.

Angolo di sfasamento	Stima Flusso destro (F_{dx})	Stima Flusso sinistro (F_{sx})	Differenza ($F_{dx} - F_{sx}$)
18	1	0.8218	0.1782
18	0.9718	1	-0.0282
18	1	0.6676	0.3324
18	1	0.7504	0.2496
18	1	0.7536	0.2464
18	0.9750	1	-0.0250
18	1	0.8898	0.1102
18	0.6216	1	-0.3484
18	1	0.9372	0.0628
18	0.6072	1	-0.3928

Tabella 4: I valori del flusso destro e sinistro, e la loro differenza, relativi ad ogni passo lungo la traiettoria diagonale



Il flusso complessivo nell'arco delle dodici prove sui lati destro e sinistro



La differenza tra flusso destro e sinistro nell'arco delle dodici prove lungo la traiettoria diagonale

Quest'ultima prova è la meno soddisfacente; lo specchio è stato spostato lungo una diagonale, partendo da un punto vicino alla parete destra e terminando in un punto in prossimità della parete sinistra.

La tabella e il grafico soprastanti riportano i risultati dell'esperimento; idealmente si sarebbero dovuti ottenere valori del flusso ottico destro inizialmente superiori a quelli del lato sinistro e la differenza fra questi due valori sarebbe dovuta diminuire progressivamente coll'avvicinarsi alla parte centrale del corridoio sino ad annullarsi in corrispondenza del centro per poi aumentare in favore dei valori del lato sinistro mano a mano che i campionamenti si facevano sempre più vicini alla parete sinistra.

Ciò che si attendeva si è riscontrato solo parzialmente, ossia: i valori del flusso ottico destro nelle prime acquisizioni sono superiori ai corrispondenti valori del lato sinistro e nelle acquisizioni centrali si osservano valori della differenza molto piccoli; un discorso analogo non può essere fatto invece per gli ultimi campioni della prova, e soprattutto la caratteristica che più manca all'esperimento per poter soddisfare le ipotesi iniziali è il passaggio progressivo da una predominanza dei valori di destra a una predominanza dei valori di sinistra.

A giustificazione di questi risultati ci sembra opportuno sottolineare che la quarta prova è stata caratterizzata da ulteriori elementi che hanno allontanato l'esperimento dall'idealità; infatti la distanza fra un'acquisizione e la successiva è aumentata sino a oltre 30 centimetri (11 campioni in circa 3 metri).

3.3 Conclusioni

In generale l'algoritmo ha prodotto una stima dei flussi ottici laterali in accordo con la teoria, sebbene nel corso di alcune prove si siano riscontrati valori di stima in contrasto con quelli attesi. In particolare le discordanze più evidenti si sono osservate utilizzando la quarta serie di dati, cioè lungo la traiettoria diagonale. Inoltre la teoria avrebbe voluto che nello svolgersi delle tre prove caratterizzate da traiettorie parallele alle pareti del corridoio, la differenza fra i valori del flusso destro e sinistro fosse costante; questo obiettivo non è stato conseguito nei modi sperati poiché osservando le rappresentazioni grafiche delle differenze del flusso si nota come le curve non siano rettilinee, bensì leggermente oscillanti. Ciò si può in gran parte attribuire ai limiti di cui soffrono i set di dati in nostro possesso, dovuti principalmente all'elevata distanza tra un'acquisizione e l'altra. Tuttavia, in considerazione del fatto che i risultati ottenuti sono stati comunque incoraggianti, si è deciso di proseguire nello studio del metodo e di implementare gli algoritmi su di un robot autonomo, in modo da poter variare a piacimento i parametri che hanno condizionato l'esito del test precedente.

Si è quindi proceduto alla riscrittura in C++ degli algoritmi in esame al fine di fare navigare un robot in tempo reale all'interno del corridoio opportunamente condizionato.

4 La realizzazione in laboratorio

4.1 Setup del test

L'implementazione pratica degli algoritmi si è svolta nel Laboratorio di Robotica Avanzata (LRA) utilizzando il robot TOBOR (Figura 13). Tale robot era collegato ad un PC Linux tramite un radio-modem, che consentiva di pilotarlo a distanza.



Figura 13: Tobor allestito per l'esperimento di navigazione

Su Tobor sono state montate diverse telecamere, tutte collegate ad Herbie tramite una scheda di acquisizione basata su chip BT848 che consentiva di ricevere il segnale video sul PC (Figura 14).

È poi stato ricostruito il corridoio di 3 metri con bande laterali colorate per gli esperimenti di navigazione, e il supporto per specchio conico e telecamera (Figura 15).

Il passo successivo è stata l'implementazione software di acquisizione, elaborazione e navigazione, per le quali si è utilizzato l'ANSI C++. TOBOR viene pilotato tramite la nota interfaccia SAPHIRA, mentre l'interfacciamento con il frame-grabber è stato reso possibile da un modulo Linux. Proprio quest'ultimo settore è meritevole di un maggiore approfondimento nella parte successiva, perché fonte di alcuni problemi e limitazioni.

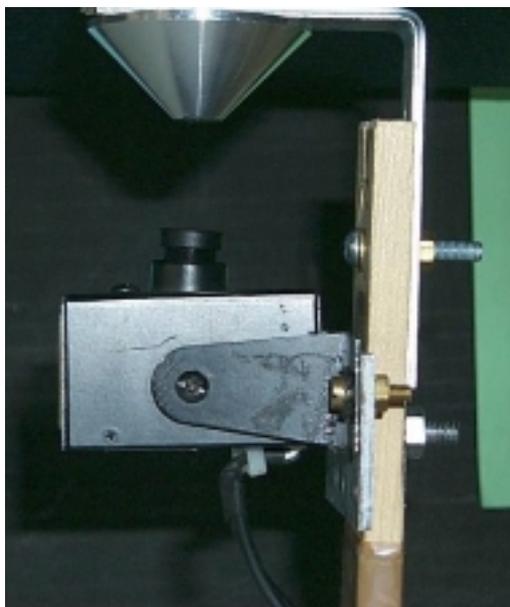


Figura 14: Lo specchio conico e la telecamera montati sui rispettivi supporti

4.2 Acquisizione

Prima di tutto occorre osservare che l'interfacciamento al frame grabber avviene tramite un modulo Linux che ha causato numerosi problemi. In particolare:

- Si è rivelato incompatibile con le librerie di stream del C++ (per intendersi, quelle le cui chiamate sono contenute in `iostream.h` e `fstream.h`).
- Si è rivelato incompatibile con alcune funzioni (ad esempio la `pow()`) per le quali è stato necessario scrivere degli *stub*.
- Ha rivelato una certa instabilità: saltuariamente si blocca causando il blocco della rete e rendendo necessario il riavvio di Herbie.

Nei primi due casi il debugger ha catturato un'eccezione nella funzione di sistema `nanowait()`, con ogni probabilità un'incompatibilità o un bug a basso livello. Risolti i problemi del frame grabber e riscritte alcune delle vecchie funzioni di linearizzazione per velocizzarle, si è provveduto a cominciare le acquisizioni con la telecamera di Tobor. Purtroppo i risultati sono stati molto deludenti: a questo proposito, in Figura 16 si vede un'acquisizione ottenuta dopo attentissima taratura della telecamera, e la relativa linearizzazione.



Figura 15: Il corridoio teatro delle prove di navigazione

Si può facilmente capire che da una linearizzazione come quella riportata è assolutamente impossibile trarre qualsivoglia informazione utile alla navigazione; purtroppo, da una immagine siffatta non è possibile ottenere risultati migliori. Infatti, anche se in apparenza l'acquisizione è di qualità accettabile, guardando con maggiore attenzione ci si accorge che:

1. Sono presenti zone ad altissima luminosità che riportano riflessi dell'ambiente circostante assolutamente ineliminabili, che per la natura dell'algoritmo di linearizzazione influenzano negativamente e in modo determinante anche i valori di zone della corona circolare relativamente distanti da esse.
2. I colori sono assai poco saturi e uniformi, presentando anzi un "effetto neve" che finisce per creare un rumore diffuso che appiattisce i valori di luminosità.

E' stato fatto ogni tentativo per cercare di rendere utilizzabile l'immagine acquisita dal frame grabber di Tobor. Si è andati dalle più svariate tarature della telecamera (luminosità, contrasto, saturazione) all'impiego di coperture delle più svariate forme e dimensioni. Si è addirittura richiusa la struttura telecamera-cono all'interno di un cilindro nero di cartone, senza tuttavia riuscire ad attenuare in alcun modo la saturazione di alcune zone del cono.

Un ulteriore tentativo è stato fatto tramite la modifica degli algoritmi di linearizzazione. In particolare:

- Si è provato a filtrare l'immagine tramite filtri integrali, passa-alto e passa-basso, e filtri più complessi ottenuti tramite matrici di convoluzione, senza tuttavia riuscire ad ottenere miglioramenti apprezzabili. Da notare comunque che l'eventuale applicazione di tali filtri a run-time avrebbe probabilmente comportato un eccessivo carico computazionale.
- Si è modificato in vari modi l'algoritmo di linearizzazione propriamente detto, allargando o stringendo la corona circolare, aumentando o diminuendo il numero di campioni per grado di circonferenza, modificando la norma applicata a tali campioni, senza tuttavia, neanche in questo caso, riuscire ad ottenere un'immagine trattabile.
- Altri tentativi sono stati fatti intervenendo sull'ambiente esterno, cambiando il tipo di illuminazione, il colore delle pareti, inserendo finte pareti nere per attenuare la luminosità generale, ma anche qui con scarsi risultati.

In conclusione, l'insieme telecamera-frame grabber di Tobor, pur superando uno dei limiti maggiori della Quickcam (il basso *frame rate*), non possiede una qualità dell'immagine neppure lontanamente sufficiente per poter stimare in maniera affidabile il CLF (*Conical Lateral Flow*). Non è escluso che, utilizzando un qualche sofisticato sistema di filtraggio, i risultati possano migliorare (a prezzo, però, di un'accresciuta complessità computazionale che potrebbe limitare decisamente il numero di elaborazioni al secondo), ma la base di partenza è così friabile che non vale probabilmente neppure la pena di tentare. Un ulteriore tentativo è stato fatto utilizzando la Quickcam,

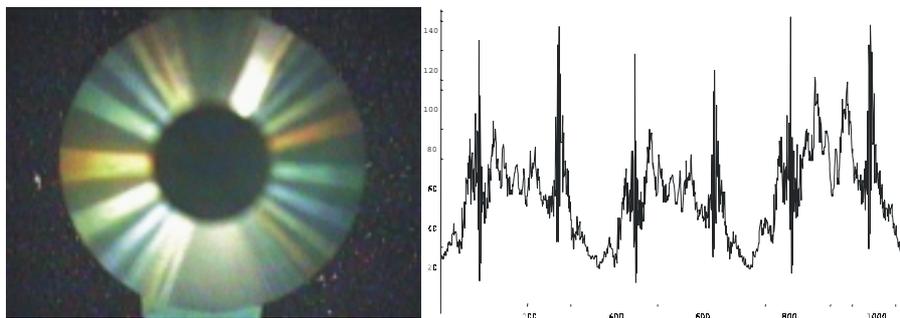


Figura 16: Un'acquisizione eseguita con il frame grabber di Tobor e la relativa linearizzazione

sfruttando una nuova libreria (la `libcqcam`) che ha consentito di ottenere prestazioni migliori rispetto alla precedente. In particolare, tale libreria consente di:

- Eliminare l'effetto "luci natalizie" (Christmas light) che è un difetto tipico della quickcam.
- Tarare automaticamente i parametri di luminosità, soprattutto il bianco.
- Ottenere un frame-rate tutto sommato decoroso (circa 1 frame ogni 0.7 secondi a 240x240x32bpp).

Un esempio delle acquisizioni fatte con la quickcam (e relative linearizzazioni) è dato in Figura 17.

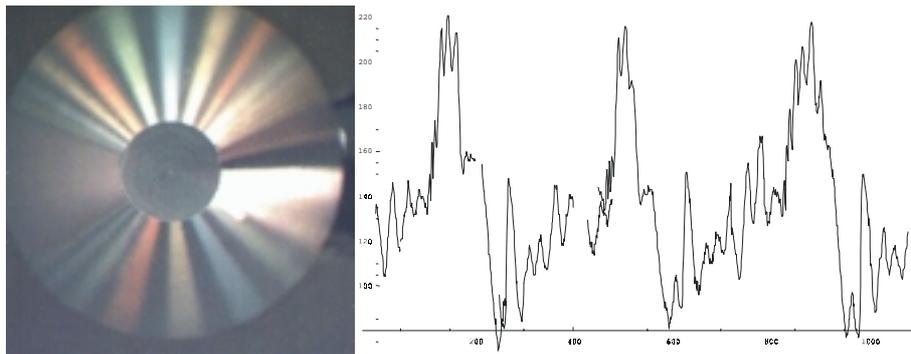


Figura 17: Un'acquisizione eseguita con la QuickCam, e la relativa linearizzazione

L'ultimo tentativo è stato fatto con una telecamera Canon UC900 di buona qualità, che ha finalmente consentito di produrre i primi risultati positivi. In Figura 18 vediamo un'immagine della telecamera con l'ormai abituale linearizzazione. Come si può notare, i picchi di luminosità sono molto più regolari e il rumore generale è molto inferiore. Ulteriore riprova della buona qualità dell'immagine è data dalla Figura 19, che rappresenta la luminosità media lungo i 360 gradi del cono: confrontandola con l'istantanea da cui è tratta, si nota la buona corrispondenza tra i picchi di luminosità della fotografia e quelli della linearizzazione.

È stato quindi possibile procedere all'esecuzione di prove di navigazione significative: si è deciso di partire con il metodo di Horn e Schunk e si è fatto muovere il robot lungo il centro del corridoio e in prossimità delle 2 pareti.

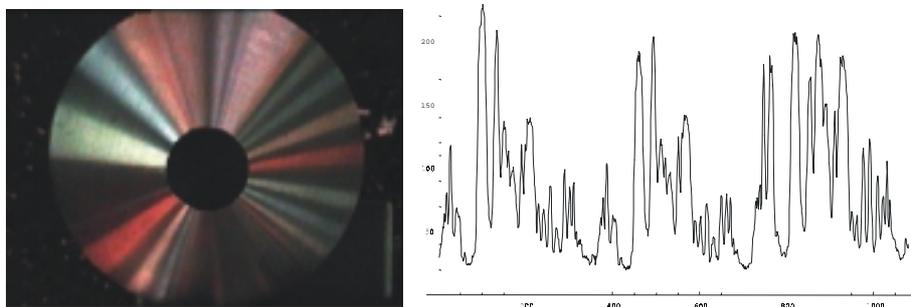


Figura 18: Un'acquisizione eseguita con la Canon UC900, e la relativa linearizzazione

Si è verificato che l'elaborazione di ogni singolo frame era controproducente (ricordiamo che la telecamera in questione inviava al frame grabber 25 schermate al secondo), in quanto fattori ambientali ostili (notevoli vibrazioni della telecamera, dondolio del supporto, piccoli cambi di direzione del robot dovuti al pavimento diseguale) portavano a stime del flusso non coerenti: si è trovato sperimentalmente che i risultati migliori si ottengono considerando un frame ogni 15-30.

Ricordiamo che il metodo in esame si basa sul calcolo della differenza settore per settore del flusso ottico lungo i lati destro e sinistro del cono, che conduce attraverso vari passaggi ad un valore compreso tra -1 e 1. Più realisticamente, ci aspettiamo di avere valori prossimi allo 0 al centro del corridoio, positivi a sinistra e negativi a destra.

A titolo rappresentativo, in Figura 20 vediamo i risultati ottenuti sull'arco di 50 acquisizioni lungo la traiettoria centrale del corridoio, mentre in Figura 21 vediamo i risultati ottenuti lungo una traiettoria vicina alla parete sinistra

A prima vista, la distribuzione dei valori appare molto irregolare, ma a una più attenta osservazione si nota come la prima distribuzione sia "centrata" intorno allo 0, mentre la seconda intorno allo 0.2. Sin dalle prime prove è apparso subito chiaro che il rumore presente nelle acquisizioni avrebbe costituito un serio problema. Tale irregolarità costituisce ovviamente un grosso limite nel caso si voglia (come si vuole) fare navigare il robot in tempo reale: è infatti impensabile affidarsi alla singola acquisizione per stabilire la posizione del robot all'interno del corridoio, in quanto abbiamo visto come tali valori, pur coerenti nella media sulla lunga distanza, oscillino anche pesantemente e in alcuni casi (a causa degli elevati errori di acquisizione) siano decisamente fuori scala. Si è quindi introdotto un sistema di filtraggio che consenta di avere, dopo le opportune tarature, una stima sempre significativa del flusso

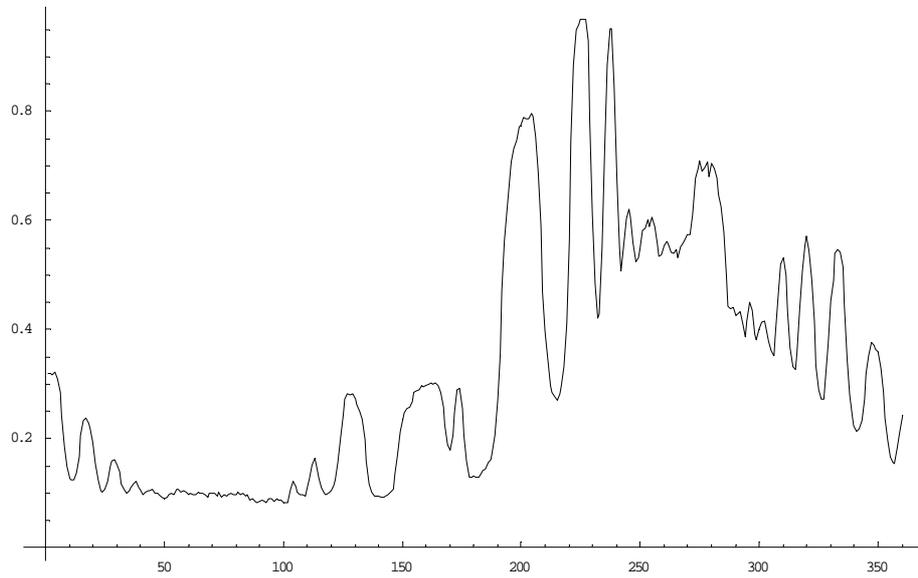


Figura 19: La luminosità media dell'immagine lungo la circonferenza del cono

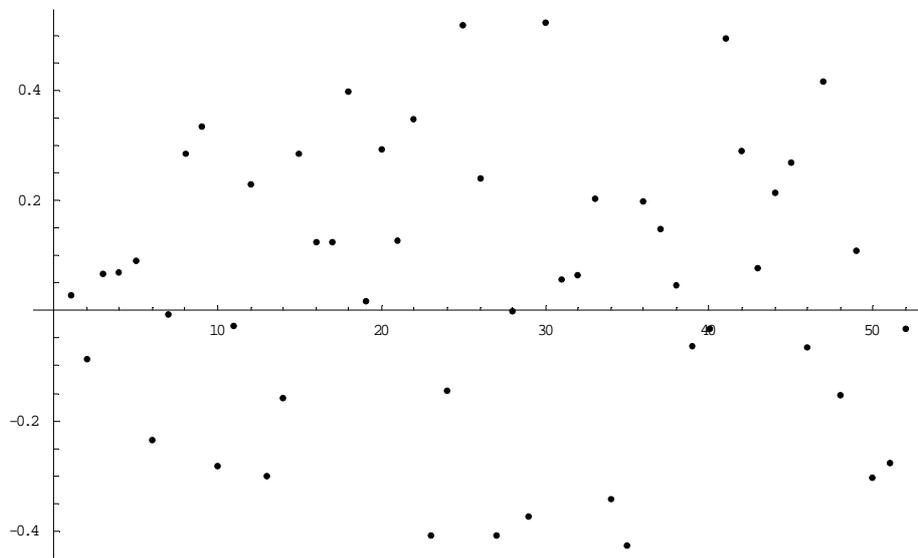


Figura 20: Quando il robot percorre il corridoio centrale, i valori del flusso sono, con buona approssimazione, distribuiti intorno allo 0

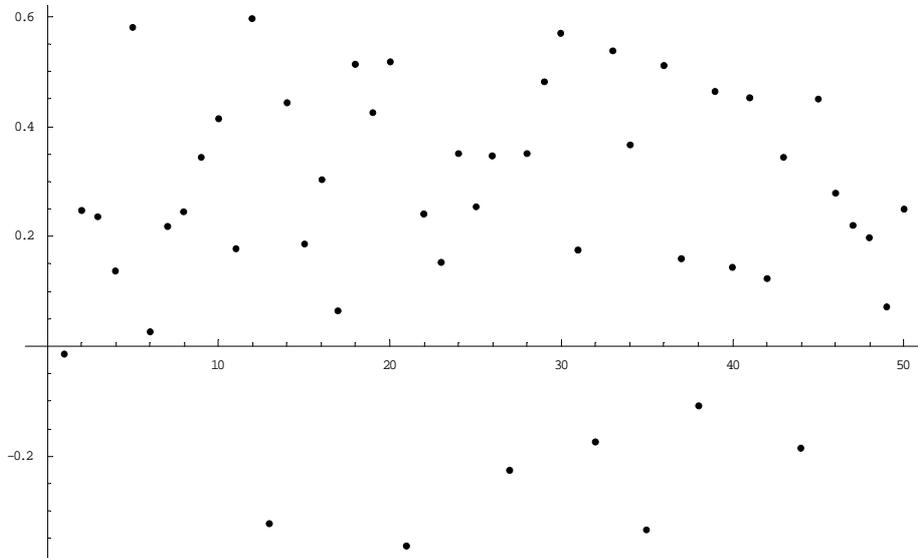
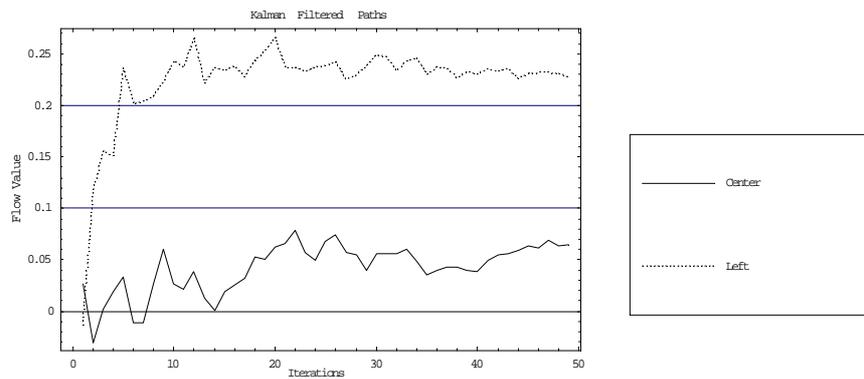


Figura 21: Quando il robot si muove in prossimità della parete sinistra, la maggior parte dei valori del flusso è superiore a 0.2

ottico. Sono stati presi in considerazione vari sistemi, partendo dai più semplici (medie aritmetiche, geometriche, minimi quadrati) e approdando infine al filtro di Kalman [23]. Come si può vedere dalla figura seguente, i risultati sono buoni.



Le due precedenti traiettorie dopo essere state opportunamente filtrate.

5 Introduzione del filtro di Kalman discreto per la compensazione degli errori di misura

Il filtro di Kalman discreto è un set di equazioni che fornisce un sistema rapido ed efficiente per stimare lo stato presente e persino futuro di un sistema lineare tempo-discreto, basandosi sulla conoscenza del suo stato passato e sulla misurazione attuale delle variabili. I punti di forza di questo metodo risiedono, oltre che nella sua efficienza, anche nella natura ricorsiva dell'algoritmo, che consente di tenere conto di tutta la storia passata del processo con un dispendio minimo in termini di tempo di calcolo[25][22].

5.1 Principi generali del filtro

Sia dato un processo che può essere descritto dall'equazione lineare stocastica:

$$x_{k+1} = A_k x_k + B_{u_k} + w_k \quad (6)$$

Ove:

- w_k è il rumore del processo con distribuzione normale

$$p(w) \sim N(0, Q) \quad (7)$$

- $A(n \times n)$ è la matrice che mette in relazione lo stato all'istante k con lo stato all'istante $k + 1$
- u è l'input.
- B è la matrice che mette in relazione u con x .

Sia poi

$$z_k = H_k x_k + v_k \quad (8)$$

la misura del processo, ove:

- v_k è il rumore di misurazione, con

$$p(v) \sim N(0, R) \quad (9)$$

- H_k è la matrice che mette in relazione lo stato del processo con la misura effettuata.

Ora supponiamo di voler misurare lo stato del processo all'istante k utilizzando due stime:

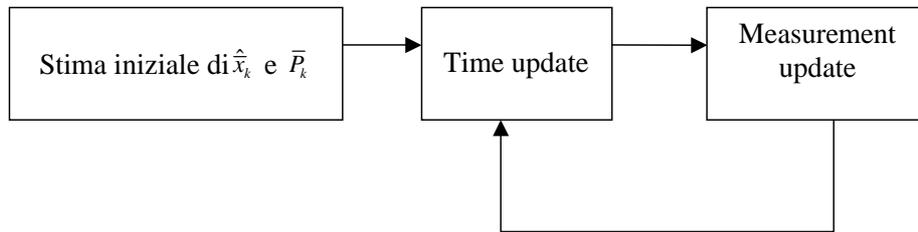


Figura 22: Lo schema di funzionamento del filtro di Kalman discreto

- \hat{x}_k è la stima *a priori*, basata sulla conoscenza dello stato del processo agli istanti che precedono k .
- \hat{x}_k è la stima *a posteriori*, basata anche sulla misura z_k dello stato del processo all'istante k .

con, rispettivamente:

- $e_k^- = x_k - \hat{x}_k$ errore *a priori*
- $e_k = x_k - \hat{x}_k$ errore *a posteriori*

ove si è utilizzato x_k per indicare lo stato *reale* del processo.

5.2 L'algoritmo del filtro

Il filtro di Kalman è basato su due passi distinti:

1. *time update*: è responsabile della *predizione*: sfruttando lo stato attuale e la covarianza dell'errore, questo passo produce una stima *a priori* dello stato nell'istante successivo.
2. *measurement update*: è responsabile della *correzione*, e sfrutta la misura attuale (operata, lo ricordiamo, dal rumore) per correggere la stima a priori e ottenere una più accurata stima *a posteriori*.

I due passi interagiscono secondo lo schema presentato in figura 22

Più in dettaglio, le queste sono le equazioni che riguardano il passo 1.:

$$\hat{x}_{k+1} = A_k \hat{x}_k + B u_k \quad (10)$$

$$\bar{P}_{K+1} = A_k P_k A_k^T + Q_k \quad (11)$$

In cui si vede come, all'istante k , si produca una stima dello stato e della covarianza dell'errore all'istante $k + 1$ basandosi *esclusivamente* sulle stime agli istanti precedenti di stato e covarianza dell'errore, e su Q_k , matrice che misura la distribuzione di probabilità dell'errore (si veda l'equazione 7). Vediamo ora le equazioni inerenti al passo 2.:

$$K_k = \frac{\bar{P}_k H_k^T}{(H_k \bar{P}_k H_k^T + R_k)} \quad (12)$$

$$\hat{x}_k = \hat{\hat{x}}_k + K(z_k - H_k \hat{\hat{x}}_k) \quad (13)$$

$$P_k = (I - K_k H_k) \bar{P}_k \quad (14)$$

Si ricorda che R_k discende dall'equazione 9. Per il calcolo di una stima *a posteriori*, il primo passo è quello di calcolare K , ossia il *guadagno*. Tale parametro merita una spiegazione più approfondita perché si rivela cruciale per il funzionamento del filtro. Prima di tutto è bene che specificare che K , così come è calcolato nell'equazione 12, è il minimo della covarianza dell'errore *a posteriori* [22][25][24]. K viene poi utilizzato nell'equazione 13 per pesare il *residuo*, ossia la quantità $z_k - H_k \hat{\hat{x}}_k$, che rappresenta la differenza tra il valore previsto della misura $H_k \hat{\hat{x}}_k$ e il valore misurato effettivamente, z_k . Ciò che vediamo dalle equazioni 12 e 13 è che più la covarianza dell'errore di misura R_k è piccola, più K tende a essere uguale a H_k^{-1} . Più precisamente:

$$\lim_{R_k \rightarrow 0} K_k = H_k^{-1}$$

e quindi, dalla 13

$$\lim_{R_k \rightarrow 0} \hat{x}_k = \hat{\hat{x}}_k + H_k^{-1} z_k - H_k \hat{\hat{x}}_k = H_k^{-1} z_k$$

Il che ha un'interpretazione semplicissima: più la covarianza dell'errore di misura è bassa, più la misura stessa diviene affidabile per la stima *a posteriori*, e quindi viene favorita rispetto alla stima *a priori*. Viceversa, dalla 12:

$$\lim_{\bar{P}_k \rightarrow 0} K_k = 0$$

e quindi, dalla 13

$$\lim_{\bar{P}_k \rightarrow 0} \hat{x}_k = \hat{\hat{x}}_k$$

ossia, più la covarianza dell'errore a priori è bassa, più la stima *a priori* è affidabile, fino al caso limite in cui l'errore è nullo e quindi stima *a priori* e quella *a posteriori* vanno a coincidere. L'ultima equazione (la 14) consente poi di calcolare la covarianza dell'errore *a posteriori*.

5.3 Cenni sulla taratura

Meritano una ulteriore precisazione i parametri R_k e Q_k : tali valori, decisivi per la taratura e il corretto funzionamento del filtro, sono spesso misurabili “fuori linea”, prima di avviare il processo vero e proprio. Ciò ha particolarmente senso per quanto riguarda R_k , l’errore di misura, in quanto è in genere possibile effettuare qualche misurazione in un contesto noto e valutare così la distribuzione di probabilità dell’errore. Il discorso inerente a Q_k è in qualche modo differente: l’errore di processo è una quantità meno deterministica che spesso viene introdotta semplicemente per inserire incertezza in modellizzazioni poco fedeli alla realtà.

5.4 Applicazione al caso monodimensionale

L’implementazione (e la taratura) di un filtro di Kalman possono essere anche estremamente complessi. Nel nostro caso, fortunatamente, la natura del problema ci è venuta in aiuto:

- Il sistema è monodimensionale.
- La quantità da stimare è la differenza tra il flusso destro e quello sinistro lungo le pareti del cono. Ebbene, per traiettorie parallele alle pareti del corridoio (le uniche che ci interessano) tale quantità è costante.

Quindi l’equazione che governa il nostro sistema diventa:

$$x_{k+1} = A_k x_k + B u_k + w_k = x_k + w_k$$

Perché $A = 1$ (lo stato è costante) e $u = 0$ (la traiettoria non cambia). mentre la nostra misura $z \in \mathfrak{R}^1$ è:

$$z_k = H_k x_k + v_k = x_k + v_k$$

Ossia $H = 1$ in quanto misuriamo direttamente lo stato. Modellizzato il sistema, vediamo le equazioni del filtro. Quelle del *time update* diventano:

$$\hat{x}_{k+1} = \hat{x}_k$$

In quanto lo stato è identico in tutti gli istanti a meno degli errori di misura, mentre per quanto riguarda la proiezione della covarianza dell’errore:

$$\bar{P}_{k+1} = P_k + Q$$

Queste invece sono le equazioni che regolano la fase di *correzione*:

$$K_k = \frac{\bar{P}_k}{\bar{P}_k + R}$$

$$\hat{x}_k = \hat{\hat{x}}_k + K(z_k - \hat{\hat{x}}_k)$$

$$P_k = (1 - K_k)\bar{P}_k$$

La realizzazione in C++ di queste equazioni non ha creato problemi.

6 Risultati dell'applicazione del filtro

L'implementazione utilizzata, pur nella sua semplicità, ci ha consentito di ottenere risultati soddisfacenti, come si può vedere dalla Figura 23, dove la curva più regolare rappresenta i campioni filtrati; si può notare che dopo 5-10 acquisizioni il filtro è già a regime e produce una stima sufficientemente accurata del flusso.

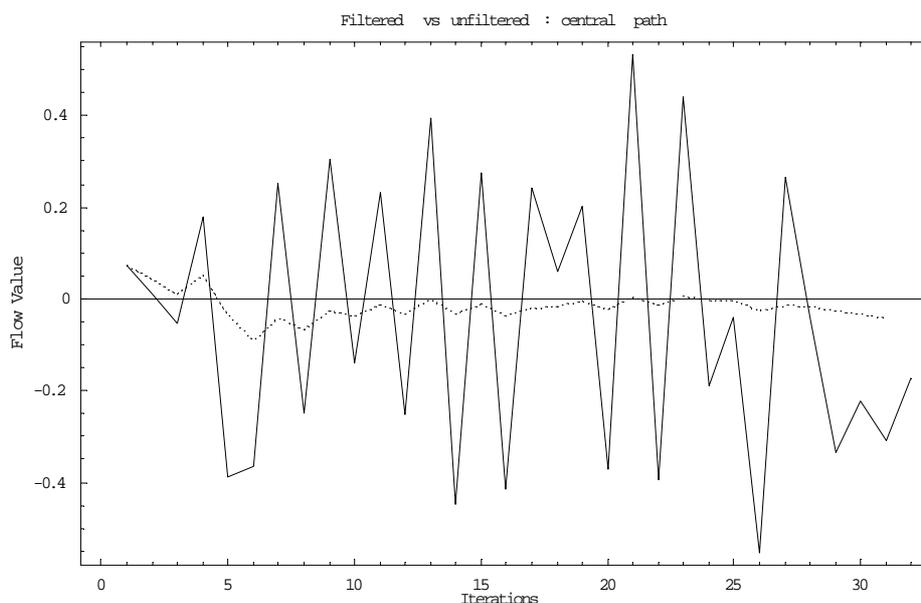


Figura 23: I valori del flusso ottico, filtrati e non filtrati, lungo la traiettoria centrale

6.1 La navigazione

Scopo della navigazione è quella di portare il robot lungo il “centro ottico” del corridoio, che non necessariamente coincide con il suo asse. L’algoritmo implementato, osservabile in Figura 24, è estremamente semplice: dopo che il filtro è giunto a regime si valuta la stima e tramite quella si cerca di stabilire la posizione all’interno del corridoio. Quindi si bloccano le acquisizioni, ci si muove in diagonale verso il centro, ci si rimette paralleli all’asse del corridoio e si riprende il ciclo fino a che il valore del flusso è all’interno di un intervallo prefissato.

La forza di questo algoritmo sta nella sua semplicità ed efficacia. Il suo principale punto debole è costituito dalla sua rigidità: il raggio di sterzata

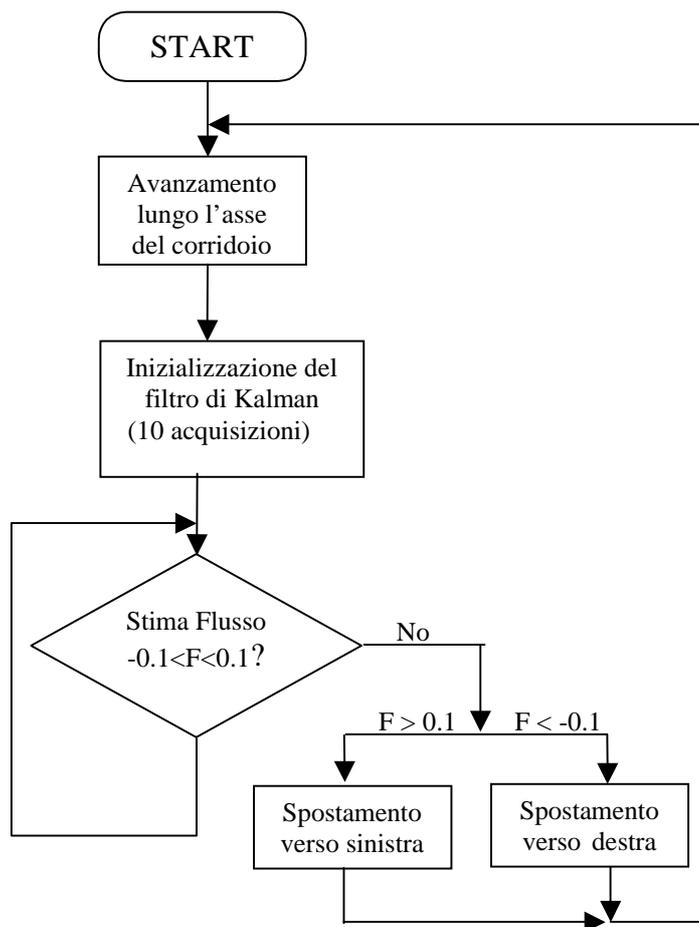


Figura 24: Il diagramma di flusso dell' algoritmo di navigazione utilizzato

e la lunghezza del tratto percorso verso il centro del corridoio sono infatti fissi, laddove con qualche accorgimento si potrebbe pensare di farli dipendere dall'entità dello "sbilanciamento" del flusso dall'una o dall'altra parte. Il problema è comunque non banale.

Ancora più difficoltoso si presenta un ulteriore perfezionamento che però consentirebbe un miglioramento decisivo delle capacità di navigazione del robot: la possibilità di continuare le acquisizioni lungo la diagonale, di continuare a stimare il flusso e di "raddrizzare" il robot al momento giusto, ossia in prossimità del centro del corridoio. Nelle fasi iniziali dell'elaborato, l'algoritmo di stima del flusso ottico era stato concepito proprio secondo questa logica; basandoci su acquisizioni effettuate staticamente lungo una diagonale, si era riusciti ad ottenere risultati abbastanza coerenti; tali risultati, nonostante i numerosi tentativi e artifici, non si sono purtroppo riscontrati nel corso delle varie esperienze di navigazione in laboratorio.

Durante queste esperienze, tutte più o meno infruttuose, il movimento diagonale ha proposto notevoli problematiche di stima del flusso, a causa della "scalatura" delle immagini dovuta al movimento verso una parete (e dal conseguente allontanamento dall'altra), oltre a richiedere un'implementazione più complessa ed elaborata del filtro di Kalman (lungo la diagonale infatti il filtro non deve più stimare una costante).

6.2 I risultati delle prove

Una volta implementato l'algoritmo di navigazione, abbiamo effettuato tre prove con il robot: esso è stato orientato parallelamente all'asse del corridoio, e nel corso delle tre esperienze, lo si è disposto prima vicino alla parete destra, quindi al centro e infine vicino alla parete sinistra. I risultati ottenuti sono stati soddisfacenti (Figura 25):

1. Nella prima esperienza il robot, partendo in prossimità della parete destra, ha iniziato il proprio movimento in avanti, per poi modificare la propria traiettoria, ruotando verso sinistra, e portandosi al centro del corridoio. A questo punto, muovendosi parallelamente all'asse del corridoio e in posizione centrale, ha proseguito la navigazione senza mai deviare. Osservando il primo grafico della Figura 25, si nota che dopo la fase di inizializzazione del filtro (10 acquisizioni), è stata rilevata una stima del flusso ottico maggiore di 0,1; a questo punto l'algoritmo di navigazione è proceduto facendo ruotare il robot verso sinistra, portandolo al centro del corridoio e ruotandolo nuovamente nella direzione dell'asse del corridoio. Terminata la nuova fase di inizializzazione del

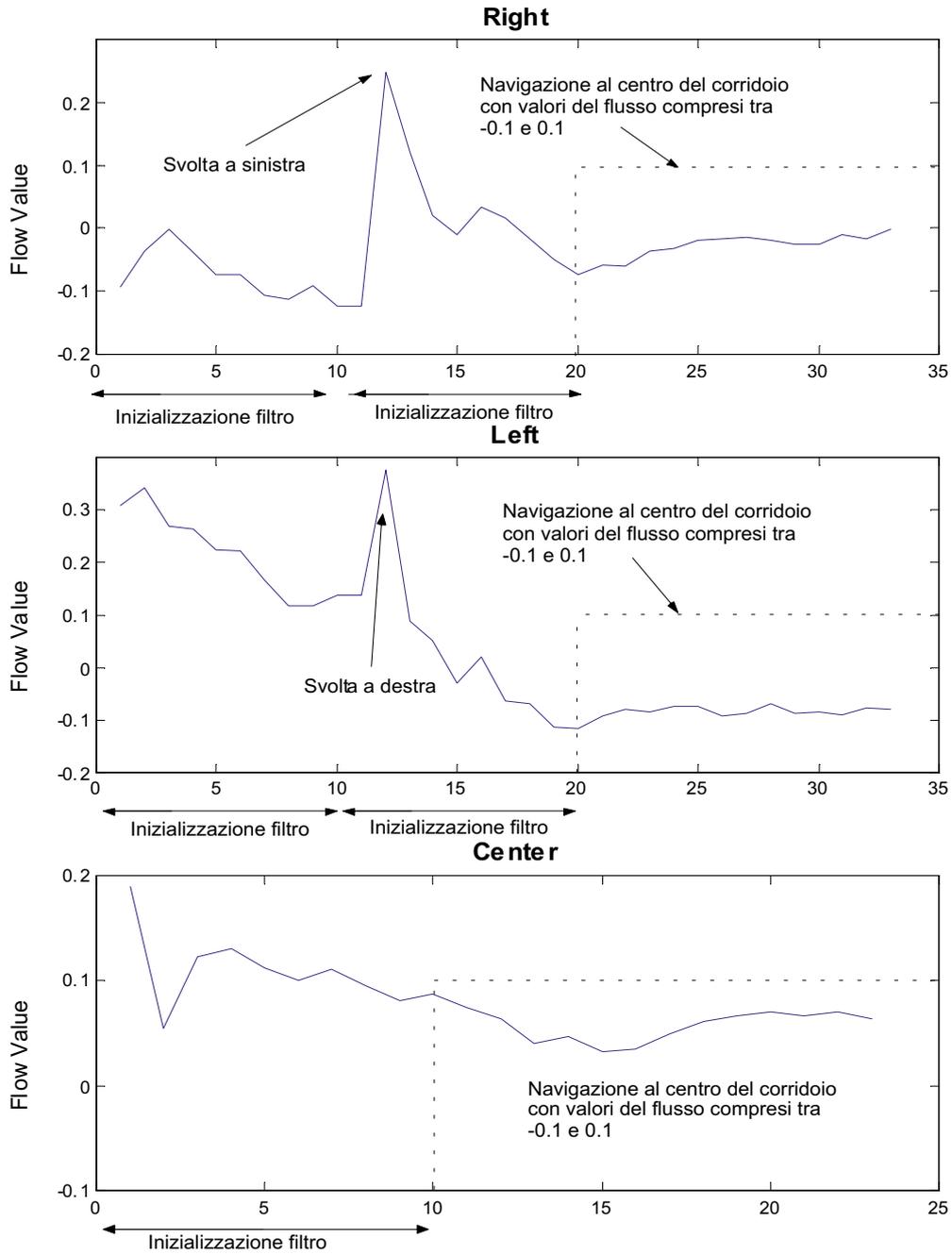


Figura 25: La navigazione del robot lungo le 3 traiettorie cardinali; a partire dall'alto: destra, sinistra e centro.

filtro (altre 10 acquisizioni) la stima del flusso si è sempre mantenuta all'interno del range fra $-0,1$ e $0,1$, consentendo quindi al robot di proseguire senza modificare la traiettoria.

2. Nella seconda esperienza, il robot, partendo al centro del corridoio, ha proseguito lungo la traiettoria iniziale, senza mai deviare. Osservando il secondo grafico della figura 4, si nota che dopo la fase di inizializzazione del filtro (10 acquisizioni), la stima del flusso si è sempre mantenuta all'interno del range fra $-0,1$ e $0,1$, consentendo quindi al robot di proseguire senza modificare la traiettoria.
3. Nella terza esperienza, il robot, partendo in prossimità della parete sinistra, ha iniziato il proprio movimento in avanti, per poi modificare la propria traiettoria, ruotando verso destra, e portandosi al centro del corridoio. A questo punto il robot, muovendosi parallelamente all'asse del corridoio e in posizione centrale, ha proseguito la navigazione senza mai deviare. Osservando il terzo grafico della Figura 22, si nota che dopo la fase di inizializzazione del filtro (10 acquisizioni), è stata rilevata una stima del flusso ottico minore di $-0,1$; a questo punto l'algoritmo di navigazione è proceduto facendo ruotare il robot verso destra, portandolo al centro del corridoio e ruotandolo nuovamente nella direzione dell'asse del corridoio. Terminata la nuova fase di inizializzazione del filtro (altre 10 acquisizioni) la stima del flusso si è sempre mantenuta all'interno del range fra $-0,1$ e $0,1$, consentendo quindi al robot di proseguire senza modificare la traiettoria.

In queste prove non è stata inserita la traiettoria diagonale. Questo perché, come ci si aspettava, nonostante diversi accorgimenti i risultati ottenuti sono del tutto insufficienti. Come già ricordato, tale traiettoria possiede delle specificità che richiedono un approccio anche teorico diverso da quello da noi adottato: da un lato le immagini non si limitano a traslare ma si riscaldano con l'avvicinamento a una parete e l'allontanamento dall'altra. Dall'altro, l'implementazione del filtro di Kalman da noi adottata stima una costante, mentre è evidente che in caso di traiettorie diagonali la differenza tra i flussi destro e sinistro varia.

6.3 Analisi dei risultati

È interessante confrontare i risultati ricavati dalle varie prove:

- Osservando le due prove in cui il robot ha iniziato la navigazione partendo in vicinanza delle pareti (prima e terza esperienza), si nota come

mentre nell'esperienza in cui il robot si è portato da sinistra sino al centro la stima del flusso dopo la seconda inizializzazione (posizione centrale) converge sempre più verso il valore 0 (valore teorico ideale indicativo del centro), nell'esperienza in cui il robot è partito vicino alla parete destra si vede come, dopo la seconda inizializzazione, la stima del flusso permane all'interno del range $-0,1 \leftrightarrow 0,1$ anche se sembra stabilizzarsi attorno al valore $-0,07$.

- Confrontando le fasi relative alla prima inizializzazione del filtro per ciascuna prova, si osserva che a differenza della prima esperienza (partenza a destra) in cui il filtro già dalle primissime acquisizioni fornisce una stima utile alla navigazione (cioè $>0,1$), nelle due restanti esperienze il filtro è andato a regime dalla 7.a acquisizione nella seconda prova (partenza dal centro) e dalla 6.a acquisizione nella terza prova (partenza da sinistra).
- Confrontando invece le fasi relative alla seconda inizializzazione del filtro (1.a e 3.a esperienza) si distingue chiaramente come nel caso della prova con partenza da sinistra (3.a esperienza) il filtro fornisce stime utili alla navigazione centrale già dopo pochissime acquisizioni, mentre nel caso con partenza da destra (1.a esperienza) le stime del flusso fornite dal filtro presentano variazioni notevoli, stabilizzate, nel range di stime utili alla navigazione centrale, solo attorno alla 9.a acquisizione.

Nell'analisi dei risultati ottenuti è necessario tenere in considerazione le condizioni dell'ambiente in cui sono state effettuate le prove, oltre ovviamente ad una serie di inevitabili imprecisioni nell'utilizzo del robot:

- L'illuminazione non uniforme nel laboratorio è stata causa di zone d'ombra lungo le pareti del corridoio (in particolare sulla parete destra).
- Le dimensioni limitate del corridoio non hanno permesso di effettuare prove diversificate al fine di ottenere stime del flusso lungo traiettorie "intermedie" (in pratica traiettorie a metà fra i lati del corridoio e il centro); una maggiore larghezza del corridoio avrebbe infatti consentito di disporre di uno spettro maggiore di stime del flusso in relazione alla posizione del robot.
- Mentre durante la fase implementativa del filtro di Kalman si è utilizzata una telecamera di ottima qualità (v. Figura 18 a pagina 37), durante la navigazione è stata impiegata una telecamera con caratteristiche decisamente inferiori. Tale cambiamento ha portato ad avere un

range di stime del flusso ottico sensibilmente ridotto; si è passati infatti da un range di ampiezza approssimativamente pari a 0,7 ($-0.35 \leftrightarrow 0.35$) ad un range di ampiezza corrispondente a circa 0,25 ($0.12 \leftrightarrow 0.12$).

- Imprecisioni, nella fase iniziale, nell'orientare manualmente il robot all'interno del corridoio, per consentirgli di seguire traiettorie parallele alle pareti.
- Imprecisioni dovute al robot durante le fasi di rotazione per l'avvicinamento alla zona centrale; infatti, al termine delle fasi "rotazione + raddrizzamento" la direzione assunta dal robot si discostava di alcuni gradi da quella dell'asse del corridoio.

7 Conclusioni

Al termine dell'esperienza di laboratorio è possibile affermare che il sistema di navigazione basato sulla stima del flusso ottico ispirata al criterio di Horn-Schunck ha fornito risultati accettabili: all'interno del corridoio condizionato, il comportamento lungo le traiettorie parallele all'asse è stato soddisfacente, mentre totalmente insufficiente è stato il comportamento lungo le traiettorie diagonali. Quest'ultimo aspetto era però previsto in considerazione sia dei risultati dell'analisi dei dati preesistenti (rif. 4.a prova) sia dell'approccio teorico da noi utilizzato.

Appare però opportuno ricordare come, mentre nella prima fase elaborativa dell'algoritmo si era giunti a buoni risultati senza la necessità di impiegare alcun sistema di filtraggio, negli esperimenti di navigazione il conseguimento di risultati positivi si è legato imprescindibilmente all'implementazione di un sistema che da un lato filtrasse in tempo reale le stime del flusso ottico, e dall'altro fornisse valori stabili al sistema di navigazione. Da qui l'importanza, decisiva nel nostro progetto, dell'applicazione del filtro di Kalman; grazie ad esso si è superata quella serie di limitazioni che impediva di realizzare esperimenti il più possibile vicini alle situazioni ideali compromettendo in modo determinante i risultati ottenuti con l'impiego del solo algoritmo di stima Horn-Schunck.

Infine, la possibilità di modificare i numerosi parametri in gioco, sia nell'implementazione dell'algoritmo di stima del flusso ottico sia nel filtro, attribuisce al sistema di navigazione quelle caratteristiche di duttilità ed elasticità indispensabili per far fronte alle diverse condizioni e situazioni di utilizzo.

A questo proposito, interessanti sviluppi futuri potrebbero essere:

- Navigazione in ambienti non condizionati (corridoi senza bande laterali, spazi aperti, etc).
- Miglioramento della capacità di navigazione sfruttando anche gli spostamenti lungo traiettorie diagonali.
- Unione con un sistema ad inseguimento di picchi significativi di luminosità.

A Coefficienti per l'errore di parallasse

C0 = 1	C1 = 1.0003	C2 = 1.0012
C3 = 1.0027	C4 = 1.0049	C5 = 1.0077
C6 = 1.0110	C7 = 1.0151	C8 = 1.0198
C9 = 1.0251	C10 = 1.0311	C11 = 1.0378
C12 = 1.0452	C13 = 1.0533	C14 = 1.0622
C15 = 1.0718	C16 = 1.0822	C17 = 1.0935
C18 = 1.1056	C19 = 1.1186	C20 = 1.1325
C21 = 1.1474	C22 = 1.1633	C23 = 1.1802
C24 = 1.1982	C25 = 1.2175	C26 = 1.2379
C27 = 1.2596	C28 = 1.2827	C29 = 1.3073
C30 = 1.3334	C31 = 1.3611	C32 = 1.3905
C33 = 1.4218	C34 = 1.4550	C35 = 1.4903
C36 = 1.5279	C37 = 1.5679	C38 = 1.6105
C39 = 1.6558	C40 = 1.7042	C41 = 1.7558
C42 = 1.8108	C43 = 1.8697	C44 = 1.9327
C45 = 2.0002	C46 = 2.0725	C47 = 2.1502
C48 = 2.2337	C49 = 2.3236	C50 = 2.4205
C51 = 2.5253	C52 = 2.6386	C53 = 2.7614
C54 = 2.8948	C55 = 3.0401	C56 = 3.1985
C57 = 3.3718	C58 = 3.5618	C59 = 3.7706
C60 = 4.0009	C61 = 4.2556	C62 = 4.5384
C63 = 4.8533	C64 = 5.2054	C65 = 5.6009
C66 = 6.0470	C67 = 6.5528	C68 = 7.1294
C69 = 7.7905	C70 = 8.5535	C71 = 9.4405
C72 = 10.4797	C73 = 11.7080	C74 = 13.1743
C75 = 14.9441	C76 = 17.1073	C77 = 19.7900
C78 = 23.1726	C79 = 27.5219	C80 = 33.2449
C81 = 40.9879	C82 = 51.8283	C83 = 67.6722
C84 = 92.1585	C85 = 132.9691	C86 = 208.7605
C87 = 375.5019	C88 = 875.7262	C89 = 4377.2974

B Principali funzioni utilizzate durante i test

Queste le funzioni più significative utilizzate durante i test:

- `dside* diffside (cside*, cside*)`;

Effettua la derivata discreta, rispetto al tempo, tra due acquisizioni successive.

- `double estimate(cside*, cside*, double*);`

Ritorna la differenza tra flusso destro e sinistro normalizzata a 1.

- `int parseopt(char* fname);`

Legge il file di opzioni e inizializza le variabili con i valori opportuni.

- `void loadfirstside();`

Inizializza il processo con la prima acquisizione.

- `void kinit(void);`

Inizializza il filtro di Kalman. Il numero di acquisizioni occorrenti può essere specificato, insieme agli altri parametri del filtro, all'interno del file di inizializzazione.

- `void mainloop();`

E' il ciclo principale del programma, ed esegue le funzioni che vanno ripetute ad ogni acquisizione (linearizzazione, calcolo del flusso, navigazione) richiamando le opportune funzioni.

- `void torgb (unsigned char *frame, unsigned char *rgb, int w, int h);`

Linearizza l'acquisizione per ottenere un vettore di valori RGB.

- `void update (double PriorEst, double PriorErr, double Measurement, //input double &PostEst, double &PostErr); //output`

Ad ogni ciclo, aggiorna lo stato del filtro di Kalman con le nuove stime.

- `void navigate (double flow);`

Basandosi sul valore corrente del flusso esegue una scelta di navigazione.

Riferimenti bibliografici

- [1] SANTOS-VICTOR, SANDINI, CUROTTO, GARIBALDI, Divergent stereo for robot navigation: learning from bees. IEEE Computer Society Conference on Computer Vision and Pattern Recongnitions, New York City, June15-18, 1993.
- [2] SANTOS-VICTOR, SANDINI, CUROTTO, GARIBALDI, Divergent stereo in autonomous navigation: from bees to robots. Int. Jour. of Computer Vision, 14, 159-177, Kluwer Academic Publishers, Boston, 1995.
- [3] CASSINIS, RIZZI, A proposal for an insect vision inspired control for autonomous robot by an omnidirectional device.
- [4] YAGI, YASHIDA, Real-time generation of environment map and obstacle avoidance using omnidirectional image sensor with conic mirror, in Proceedings of the IEEE Conference on Computer Vision & Pattern, pp. 160-165, 1991
- [5] HORN, SCHUNCK, Determining optical flow, artificial intelligence 17:185-204, 1981
- [6] BARRON, FLEET, BEAUCHEMIN, Systems and experiment performance of optical flow techniques, Int. Jour. of Computer Vision, 12:1, 43-77, 1994
- [7] NEGAHDARIPOUR, S. AND LEE, S. (1. December 1992). Motion recovery from image sequences using only first order optical flow information. International Journal of Computer Vision, 9(3):163-184.
- [8] GOLLAND, Use of Color for Optical Flow Estimation, Research Thesis, Israel Institute of Technology, May 1995
- [9] COLOMBO, Visione Artificiale Dinamica e Interazione Robot-Ambiente, Dottorat
- [10] BERGHOLM, CARLSSON, A theory of optical flow. Computer Vision, Graphics and Image Processing: Image Understanding, 53(2):171-188, Marzo 1991
- [11] BEUCHEMIN ARRON, The computation of optical flow. ACM Computing Surveys, 27(3):433-467, Settembre 1995.

- [12] BRANDT, Improved accuracy in gradient-based optical flow estimation. *International Journal of Computer Vision*, 25(1):5-22, 1997.
- [13] CAMPANI, VERRI, Motion analysis from first-order properties of optical flow. *Computer Vision, Graphics and Image Processing: Image Understanding*, 56(1):90-107. 1992.
- [14] CHIH-HO, CAIMI, Determination of horizontal motion through optical flow computations. *Journal of Computer Science and Technology*, 12(2):133-144, 1997.
- [15] DEL BIMBO, NESI, SANZ, Analysis of optical flow constraints. *IEEE Transactions on Image Processing*, 4(4):460-469, 1995.
- [16] HORN, SCHUNCK, 'Determining optical flow': a retrospective. *Artificial Intelligence*, 59(1-2):81-87, 1 Gennaio 1981.
- [17] HORN, SCHUNCK, 'Determining optical flow': a retrospective. *Artificial Intelligence*, 59(1-2):81-87, 1 Gennaio 1993
- [18] KANATANI, Transformation of optical flow by camera rotation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(2):131-143, Marzo 1988.
- [19] NEUMANN, *Optical flow Computers and Graphics*, 18(1):17-19, 1984.
- [20] SCHUNCK, The Image Flow Constraint Equation. *Computer Vision, Graphics and Image Processing*, 35(1):20-46, Luglio 1986.
- [21] VERRI, GIROSI, TORRE, Differential Techniques for Optical Flow. *J. of the Optical Society of America A*, 7(5):912-922, 1990.
- [22] MAYBECK, *Stochastic Models, Estimation, and Control, Volume 1*, Academic Press, Inc., 1979
- [23] KALMAN, A New Approach to Linear Filtering and Prediction Problems, *Transaction of the ASME Journal of Basic Engineering*, pp. 35-45, Marzo 1960.
- [24] JULIER, UHLMAN, "A General Method of Approximating Nonlinear Transformations of Probability Distributions", Robotics Re-search Group, Department of Engineering Science, University of Oxford [citato 14 November 1995]. Available from <http://www.robots.ox.ac.uk/~si-ju/work/publications/Unscented.zip>.

Vedi anche: “A New Approach for Filtering Nonlinear Systems” by S. J. Julier, J. K. Uhlmann, and H. F. Durrant-Whyte, Proceedings of the 1995 American Control Conference, Seattle, Washington, Pagine:1628-1632, disponibile su http://www.robots.ox.ac.uk/~siju/work/publications/ACC95_pr.zip.

Vedi anche l'home page di Simon Julier <http://www.robots.ox.ac.uk/~siju/>.

- [25] WELCH, BISHOP, An introduction to the Kalman filter, <http://www.cs.unc.edu/~welch>