

ALMA MATER STUDIORUM · UNIVERSITÀ DI BOLOGNA

SCUOLA DI SCIENZE

Corso di Laurea in Matematica

**PUNTI IMPROPRI
NELLA ROBOTICA MOBILE**

Tesi di Laurea in Geometria

Relatore:
Chiar.mo Prof.
MASSIMO FERRI

Presentata da:
ELISA AMADORI

Sessione III
Anno Accademico 2014/2015

“Alla mia famiglia.”

Indice

Introduzione	2
1 Nozioni preliminari	4
2 Utilizzo dei punti di fuga nella navigazione interna	8
2.1 Punti di fuga per ricavare la posa del robot	8
2.2 Punti di fuga per sterzare un robot	9
2.3 Punti di fuga per la calibrazione della telecamera	9
3 Stima della posa del robot e metodi impiegati	11
3.1 Trasformata di Hough e raggruppamento K-means	11
3.2 Navigazione esterna	25
3.3 Attendibilità delle fonti	30
Conclusione	33
Bibliografia	34

Introduzione

Negli ultimi decenni molti autori hanno affrontato varie sfide per quanto riguarda la navigazione autonoma di robot e sono state proposte diverse soluzioni per superare le difficoltà di piattaforme di navigazioni intelligenti. I campi di applicazione principali riguardano la navigazione di un robot in un ambiente esterno in cui si cerca di ottenere una descrizione affidabile dell'ambiente grazie all'estrazione di caratteristiche della scena stradale come linee di demarcazione, di confine e altre, che poi permettono di rilevare un modello affidabile della strada su cui il robot può viaggiare. Esistono diversi approcci a seconda se si ha a che fare con strade ben progettate come autostrade che sono pavimentate e ricche di contorni adatte per il rilevamento dei bordi, oppure ambienti strutturati urbani dove il riconoscimento automatico della segnaletica stradale è un problema chiave all'interno del dominio della guida autonoma, oppure ancora ambienti non strutturati in cui non è permessa nessuna ipotesi permanente sulla struttura della strada. Molto importante è la navigazione in un ambiente interno, come corridoi o spazi chiusi di un edificio in cui si studia il problema del recupero della disposizione spaziale di una scena basato sull'estrazione di linee verticali e orizzontali estratte dall'ambiente, in combinazione con un insieme di vincoli geometrici, che permettono di rilevare i confini pavimento-pareti-soffitto e un aspetto ancora più rilevante è quello di far orientare e guidare il robot negli interni dell'edificio, anche nei casi di ostacoli sconosciuti che occludono la visione.

Un robot può identificare facciate degli edifici nel suo processo di localizzazione e mappatura, che a sua volta può essere utilizzato per aiutare la navigazione del robot stesso. Legato a tutto questo c'è il problema del recupero dell'orientamento

relativo della telecamera inserita sul robot rispetto alla scena e quindi il problema riguardo la calibrazione della telecamera.

Questo elaborato ha lo scopo di ricercare quelli che sono gli obiettivi principali della navigazione di robot mobili trattati da diversi autori e di approfondirne uno in particolare. Una delle principali sfide nella guida di un veicolo o di un robot è il rilevamento della rotta e infatti il nostro interesse si è soffermato sulla stima della posa del robot. Il rilevamento del punto di fuga gioca un ruolo fondamentale in questo campo e la maggior parte dei metodi proposti si basa su di esso. Siamo andati ad analizzare le tecniche che stimassero la posizione del robot nell'ambiente e presentiamo in particolare un metodo che vuole risalire al punto di fuga grazie all'estrazione di linee all'interno di corridoi utilizzando un approccio basato sulla trasformata di Hough e sul raggruppamento K-means. La navigazione di veicoli autonomi non riguarda solo gli ambienti interni ma presentiamo anche una descrizione generale di alcuni aspetti della navigazione su strade e su ambienti pedonali.

Presentiamo ora una breve descrizione dei capitoli. Nel primo capitolo daremo un quadro generale su alcune definizioni tra cui quella del punto di fuga in termini geometrici che ci permetterà di comprendere meglio le metodologie illustrate nei capitoli successivi. Nel secondo capitolo descriveremo alcuni obiettivi estrapolati dalla letteratura in cui mostreremo l'utilizzo dei punti di fuga per diversi ambiti quali il rilevamento della posa del robot, lo sterzo del robot e la calibrazione della telecamera. Approfondiremo nel terzo capitolo lo studio della posa di un robot in primo luogo in ambienti interni grazie al metodo citato sopra sul raggruppamento K-means e in secondo luogo tratteremo alcune osservazioni che riguardano la navigazione esterna. Infine nella parte finale puntualizzeremo alcuni concetti e in particolare presenteremo una sottolineatura riscontrata durante il lavoro di questo elaborato.

Capitolo 1

Nozioni preliminari

In questo capitolo introduciamo alcune nozioni fondamentali sulla teoria dei punti impropri e punti di fuga che ci permetteranno di trattare alcuni aspetti della navigazione interna di robot in movimento.

Definizione 1.1. (Spazio proiettivo) Diremo *spazio proiettivo* sul campo K ogni terna (V, P, θ) , dove V è uno spazio vettoriale su K , P è un insieme e $\theta : V \setminus \{0\} \rightarrow P$ è un'applicazione suriettiva tale che

$$\theta(v) = \theta(w) \Leftrightarrow \exists \alpha \in K, \quad w = \alpha v.$$

Se V ha dimensione finita $n+1$, diciamo che lo spazio proiettivo ha dimensione n . Gli elementi del sostegno P si dicono punti dello spazio proiettivo.

Solitamente si indicherà uno spazio proiettivo con il simbolo del suo sostegno P .

Per ogni spazio vettoriale non banale V su K c'è uno spazio proiettivo naturalmente associato $(V, P(V), \theta)$, dove, detta \sim la relazione di proporzionalità su $V \setminus \{0\}$, $v \sim w \Leftrightarrow \exists \alpha \in K, w = \alpha v$, si definisce $P(V) = V \setminus \{0\} / \sim$ e

$$\theta : V \setminus \{0\} \rightarrow P(V)$$

$$v \mapsto [v]$$

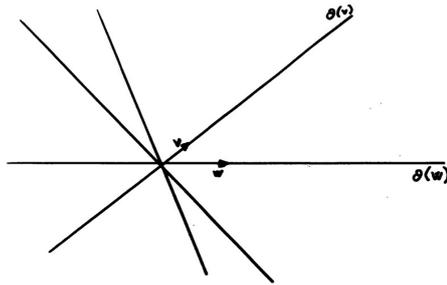
è la proiezione canonica.

Esempio 1.1

$V = \{\text{segmenti di un piano } \pi, \text{ uscenti da un punto } Q\}$

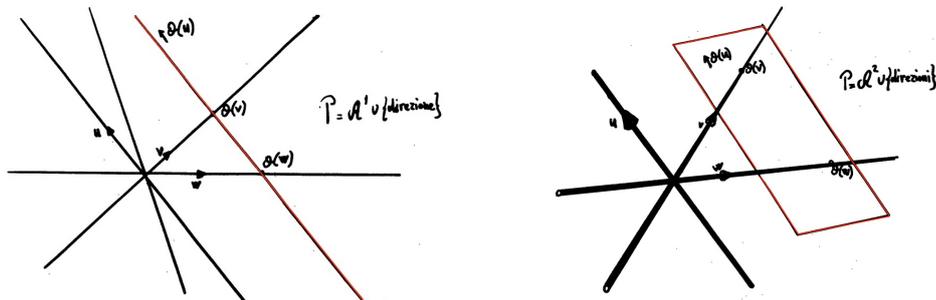
$P = \text{rette di } \pi \text{ passanti per } Q$

Dimensione=1

**Esempio 1.2**

$V = K^{n+1}$

$P = A^n \cup \{\text{direzioni di } A^n\}$ (A^n spazio affine n-dimensionale)



Definizione 1.2. (Proiettività) Siano (V, P, θ) e (V', P', θ') spazi proiettivi su K . Sia $T : V \rightarrow V'$ una trasformazione lineare iniettiva. Allora T induce un'applicazione $T' : V \setminus \{0\} \rightarrow V' \setminus \{0\}$ compatibile con le proporzionalità, cioè tale che $u \sim v \Leftrightarrow T'(u) = T'(v)$. Dunque T induce anche un'applicazione

$$w_T : P \rightarrow P'$$

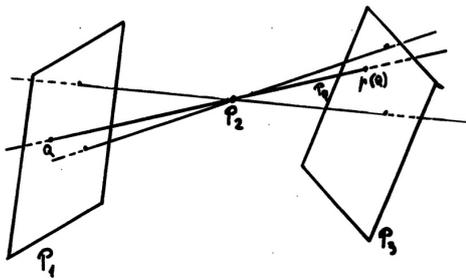
$$\theta(v) \mapsto \theta'(T(v))$$

che verrà detta associata a T .

Chiameremo *proiettività* da P a P' ogni applicazione $w : P \rightarrow P'$ tale che esista una trasformazione lineare iniettiva T per cui sia $w = w_T$. Una proiettività da P a se stesso viene chiamata *omografia*.

Definizione 1.3. (Prospettività) Dati sottospazi P' e P'' di uno spazio proiettivo P , chiamiamo *sottospazio intersezione* [congiungente] il sottospazio proiettivo associato al sottospazio vettoriale intersezione [congiungente] dei sottospazi vettoriali ad essi associati.

Siano P_1, P_2, P_3 sottospazi di uno spazio proiettivo P . Per ogni punto $Q \in P_1$, sia P_Q il sottospazio congiungente di Q e P_2 . Se per ogni $Q \in P_1$ vale che $P_Q \cap P_3$ è costituito da un solo punto, l'applicazione $p : P_1 \rightarrow P_3$ che a Q associa tale punto viene detta *proiezione* di P_1 da P_2 su P_3 .



Se $\dim P_1 = \dim P_3$ e la proiezione non è costante, essa viene detta *prospettività* e P_2 è detto *spazio di prospettiva*.

Teorema 1.0.1. *Ogni proiettività fra sottospazi di uguale dimensione è la composizione di un numero finito di prospettività.*

Ora vediamo che collegamento sussiste tra spazio affine e spazio proiettivo.

Definizione 1.4. (Ampliamento proiettivo di uno spazio affine) Sia A^n uno spazio affine n -dimensionale su K ; sia inoltre \vec{A}^n il suo spazio di vettori liberi. Sia infine \vec{P}^{n-1} lo spazio proiettivo naturalmente associato ad \vec{A}^n .

Proposizione 1.0.2. *I punti di \vec{P}^{n-1} sono le direzioni di A^n .*

Definiamo l'applicazione biettiva

$$\theta : K^{n+1} \setminus \{0\} \rightarrow A^n \cup \vec{P}^{n-1}$$

$$(X_0, X_1, \dots, X_n) \mapsto \begin{cases} P \equiv \left(\frac{X_1}{X_0}, \dots, \frac{X_n}{X_0} \right) & \text{se } X_0 \neq 0 \\ [\vec{u}] & \text{con } \vec{u} \equiv (X_1, \dots, X_n) \text{ se } X_0 = 0 \end{cases}$$

Proposizione 1.0.3. $(K^{n+1}, A^n \cup \vec{P}^{n-1}, \theta)$ è uno spazio proiettivo. Tale spazio è detto *ampliamento proiettivo di A^n* .

Definizione 1.5. (Punto improprio) Dato l'ampliamento proiettivo $P^n = A^n \cup \vec{P}^{n-1}$ di uno spazio affine A^n , si dice *improprio* ogni punto, sottospazio proiettivo, sottoinsieme di \vec{P}^{n-1} , che verrà chiamato *iperpiano improprio*. Un analogo elemento verrà detto *proprio* se non è contenuto nell'iperpiano improprio.

Si noti che ogni riferimento affine A^n determina un riferimento proiettivo dell'ampliamento proiettivo, tale che dette x_1, \dots, x_n le coordinate affini di un punto $P \in A^n$ e X_0, X_1, \dots, X_n le coordinate proiettive di $\theta^{-1}(P)$ per $i = 1, \dots, n$ valga $x_i = \frac{X_i}{X_0}$.

Con tali riferimenti vale anche $X_0, X_1, \dots, X_n = \lambda(1, x_1, \dots, x_n)$.

Proposizione 1.0.4. Se un iperpiano di A^n ha equazione $a_1x_1 + \dots + a_nx_n + b = 0$ allora il suo ampliamento proiettivo ha equazione $bX_0 + a_1X_1 + \dots + a_nX_n = 0$.

Si consideri la proiezione π dell'iperpiano improprio \vec{P}^{n-1} su un iperpiano proprio P^{n-1} , da un punto P esterno ad entrambi. Ogni punto improprio Q_∞ viene proiettato in un punto $\pi(Q_\infty)$ di P^{n-1} ; se tale punto è proprio esso viene detto punto di fuga della corrispondente direzione rispetto a π .

Proposizione 1.0.5. Due rette affini parallele si intersecano in un punto improprio. Le loro proiezioni su un iperpiano proprio si intersecano nel punto di fuga della direzione comune.

Capitolo 2

Utilizzo dei punti di fuga nella navigazione interna

Il rilevamento del punto di fuga svolge un ruolo di primo piano in vari ambiti di visione artificiale come la navigazione di veicoli autonomi, la calibrazione della telecamera, la rettifica delle immagini e tante altre.

2.1 Punti di fuga per ricavare la posa del robot

Una delle principali sfide nel guidare un veicolo o un robot consiste nell'individuare la direzione di testa appropriata. Nella definizione di posa si includono due concetti, quello relativo alla posizione e quindi la posa è identificata dalle coordinate x e y rispetto a un sistema di riferimento, e quello relativo all'orientamento rappresentato da un angolo θ . L'utilizzo del punto di fuga è una delle tecniche più utilizzate per la navigazione interna basata sulla visione. Nel caso di ambienti artificiali vi sono un sacco di regolarità, come la presenza di gruppi di linee e piani paralleli e ortogonali, in particolare molte linee parallele sono presenti nei corridoi, come i bordi delle pareti laterali con il pavimento e il soffitto. La rotta (direzione) del robot può essere stimata grazie all'utilizzo del punto di fuga determinato da queste righe e grazie alle linee nel piano immagine. Approfondiremo questo aspetto nel capitolo successivo in cui proporremo alcuni differenti approc-

ci, in particolare l'utilizzo della trasformata di Hough e l'algoritmo di partizione (clustering) K-means [5].

2.2 Punti di fuga per sterzare un robot

I punti di fuga hanno un compito importante per la sterzata di un robot mobile. Essi servono da riferimento per la guida del veicolo e per stimarli vengono utilizzate linee parallele nell'ambiente del robot stesso. Secondo alcuni metodi [17] si procede eseguendo tre sottoattività: individuazione di linee rette, calcolo dei punti di fuga e sterzo del robot mediante punti di fuga. Essi forniscono un riferimento per rilevare la rotazione del robot. Questo è vero non solo per le linee parallele al soffitto, ma in generale per ogni coppia di linee parallele in ambienti interni o esterni. I punti di fuga sono molto utili per rilevare un accurato orientamento del robot ma hanno delle limitazioni intrinseche nelle loro capacità di rilevamento. Essi non sono sensibili allo spostamento che impone un serio problema in navigazione. Il robot può essere spostato verso una parete o un ostacolo, e il punto di fuga sarebbe ancora nello stesso posto. Quindi questo tipo di approccio non è in grado di rilevare la posizione del robot.

2.3 Punti di fuga per la calibrazione della telecamera

Nella navigazione interna i veicoli autonomi in movimento sono muniti di telecamere che consentono di rilevare diverse condizioni della scena circostante, come ad esempio eventuali ostacoli, la posizione del veicolo grazie a punti di riferimento e quindi il conseguente spostamento. Sorge, così, il problema della calibrazione della telecamera per il recupero dell'orientamento della fotocamera e ciò ha una certa rilevanza per applicazioni che richiedono conoscenze nel contesto di geometria. Le applicazioni utilizzate in una tale situazione hanno la necessità di rilevare la posizione della telecamera rispetto al sistema di riferimento della scena. La precisione desiderata varia in base all'applicazione. Nel caso di ambienti artificiali

sia interni che esterni, caratterizzati da linee orizzontali e verticali, l'orientamento relativo può essere individuato dai punti di fuga e dalle linee nel piano immagine. Il problema della calibrazione della telecamera consiste nel determinare gli elementi che regolano la relazione tra l'immagine 2-D che una telecamera percepisce e le informazioni 3-D dell'oggetto ripreso. Esso è identico al problema della posizione del robot nello spazio 3-D. Nel corso degli anni sono state applicate diverse tecniche per risolvere il problema della calibrazione di telecamere piantate sui veicoli autonomi. Alcuni metodi si sono basati sull'utilizzo di un bersaglio a forma di diamante posto sulla parete per determinare la posizione 2-D della telecamera rispetto al diamante (con l'asse ottico della telecamera passante per il centro del bersaglio-il centro della lente della fotocamera deve essere alto come il centro del diamante), oppure l'utilizzo di angoli delle case come bersagli di calibrazione. In altri casi il rilevamento della posizione della fotocamera e i parametri di orientamento sono stati ottenuti calcolando le altezze dei raggi dal centro della lente della fotocamera per i punti di controllo del piano dell'immagine, oppure basandosi su un rapporto intuitivo tra i parametri della telecamera e le linee di fuga. I parametri tarati includono la posizione, l'orientamento e la lunghezza focale di una fotocamera. I parametri intrinseci della telecamera possono essere ottenuti anche dai coefficienti di due sezioni di un cono, le quali sono percorse da un punto di fuga durante il movimento della fotocamera con spostamento arbitrario e asse fisso di rotazione [3]. Un altro approccio per la calibrazione della fotocamera, basata sull'uso della linea di fuga, consiste nel considerare una immagine monoculare di una forma esagonale [21]. Oltre ad essere in grado di calcolare la posizione della telecamera, si possono ottenere gli angoli di visualizzazione e la lunghezza focale.

Capitolo 3

Stima della posa del robot e metodi impiegati

Ci sono diversi metodi che trattano nel dettaglio lo studio dei punti di fuga per risalire alla stima della posa del robot. In questo capitolo ne proponiamo uno che abbiamo approfondito riguardante la navigazione interna e faremo brevi accenni ad un altro metodo sotto una diversa luce. Presentiamo anche brevi considerazioni per quanto riguarda la navigazione esterna.

3.1 Trasformata di Hough e raggruppamento K-means

Negli ultimi decenni molti autori hanno affrontato il problema della navigazione e il rilevamento del punto di fuga basati sull'informazione visiva e sull'algoritmo di partizione K-means, e hanno adottato metodi che rilevassero il punto di fuga grazie all'intersezione di linee parallele o orizzontali, come quelle che si trovano in ambienti interni. Il metodo che presentiamo è un metodo che non richiede questo tipo di intersezione né nel piano immagine, né nello spazio di Hough — lo introdurremo tra poco — ma si basa su un approccio che riduce la complessità e il tempo di elaborazione dell'algoritmo in larga misura. Presentiamo ora l'algoritmo che andremo ad analizzare:

1. Ottenere una nuova immagine
2. Ridimensionare l'immagine
3. Convertire da RGB a grigio
4. Eseguire l'equalizzazione dell'istogramma
5. Applicare il filtro a maschera a 45°
6. Applicare l'algoritmo di Canny
7. Eseguire la trasformata di Hough
8. Estrarre i picchi di Hough
9. Estrarre le linee di Hough
10. Determinare i punti di inizio e i punti di fine
11. Applicare il raggruppamento K-means sul set di dati (punti iniziali e punti finali)
12. Estrazione di quattro baricentri di gruppi
13. Mettere questi quattro baricentri di gruppi in una nuova matrice
14. Applicare il raggruppamento K-means
15. Estrazione del baricentro di un gruppo
16. Convalidazione del punto
17. Registrare il punto di fuga
18. Correggere la direzione

Per prima cosa diamo una breve descrizione del contesto in cui siamo in termini geometrici: le linee parallele in uno spazio tridimensionale (3D) sono proiettate come linee convergenti nel piano dell'immagine. Il punto comune di intersezione è chiamato punto di fuga e può eventualmente appartenere alla linea all'infinito del piano dell'immagine nel caso in cui le linee 3D sono parallele al piano dell'immagine. La concomitanza nel piano immagine non implica necessariamente il parallelismo in 3D (implica solo che tutte le linee 3D intersecano la linea definita dal punto focale e dal punto di fuga) e il problema di trovare linee parallele in 3D si riduce a trovare punti di fuga nel piano immagine (Fig. 3.1).

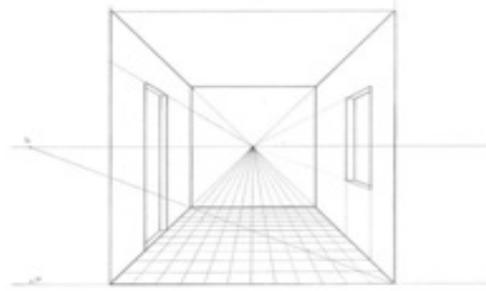


Figura 3.1: Rappresentazione del punto di fuga

Ora prendiamo come ambiente interno un corridoio e vi posizioniamo un robot dotato di un'unica fotocamera come sensore di rilevamento. Non è necessario che la fotocamera sia calibrata. Quello che vogliamo fare è andare a modellare la variazione di linee rette nel corridoio utilizzando la trasformata di Hough e il raggruppamento K-means per risalire al punto di fuga che funge da correttore di rotta del nostro robot e che gli permette di navigare con precisione all'interno del corridoio.

Per cominciare ci interessa recuperare una nuova immagine dell'ambiente che non sia modificata da eventi esterni visivi come l'illuminazione e altri eventi dannosi che influenzano la precisione del rilevamento e contribuiscono a una comprensione deficitaria dell'immagine. Questa difficoltà può essere affrontata riducendo la dipendenza del nostro algoritmo dall'accuratezza nel rilevare i bordi, e cioè consideriamo gruppi di linee invece che affidarci all'esatta posizione di singole linee come è stato fatto da altri autori per stimare il punto di intersezione. L'algoritmo proposto scarta tutte le informazioni inutili come ad esempio il numero di bordi che non sono richiesti (cioè linee con orientamento diverso da 45°) e in questo modo si risparmia più tempo soprattutto in ambienti molto disordinati. In particolare una tecnica adottata è quella di convertire l'immagine RGB in scala di grigi e poi si esegue l'equalizzazione dell'istogramma sull'immagine a scala di grigi (Fig. 3.2a e 3.2b).



(a) Immagine del corridoio scelto in scala di grigi.



(b) Risultato della scala di grigi e dell'equalizzazione dell'istogramma applicata su Fig. 3.2a.

Figura 3.2

È stato poi applicato il filtro a 45° sull'immagine prima di rilevare i bordi perché in questo modo solo alcune serie di bordi devono essere importate allo spazio di Hough, che automaticamente diminuisce il tempo di elaborazione (Fig. 3.3).

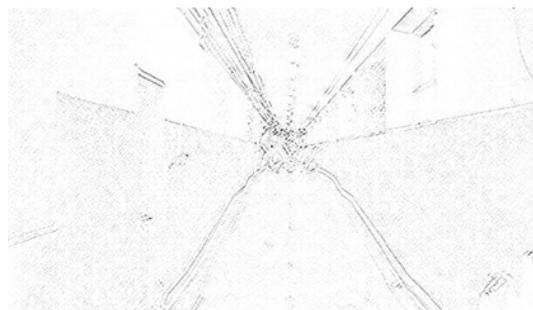


Figura 3.3: Risultato dell'applicazione del filtro a 45° sulla Fig. 3.2b

Le linee con 45° di orientamento sono caratteristiche molto affidabili.

La trasformata di Hough standard si basa sul riconoscimento delle linee di un'immagine, infatti è stata utilizzata per estrarre le linee desiderate dai bordi rilevati. Generalmente, si utilizza la rappresentazione parametrica di una retta $\rho = x * \cos(\theta) + y * \sin(\theta)$ (Fig. 3.4). La variabile ρ è la distanza dall'origine alla retta lungo un vettore perpendicolare alla retta. θ è l'angolo tra l'asse x e questo vettore. La trasformata di Hough genera un matrice di spazio parametrico le cui righe e colonne corrispondono rispettivamente a questi valori ρ e θ . Si vuole decomporre un'immagine o un set di simboli — che potrebbero essere pixel, punti

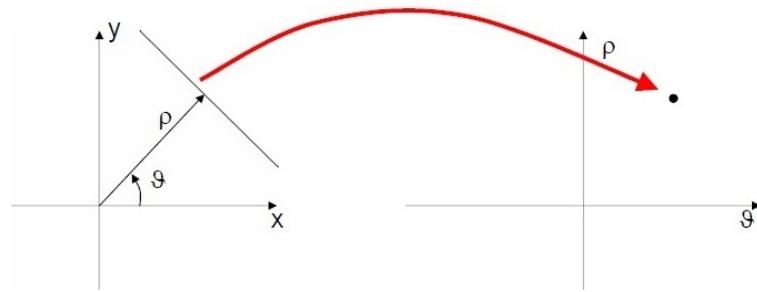
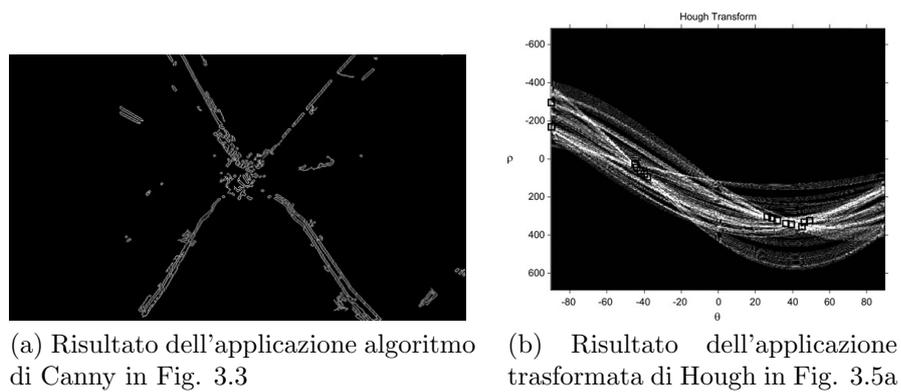


Figura 3.4: Trasformata di Hough

isolati, serie di punti di frontiera, ecc — in componenti che appartengono ad una o all'altra famiglia semplice. Nel nostro caso, abbiamo una serie di punti di frontiera e vogliamo stabilire quali linee vi si adattano meglio. La Fig. 3.5b mostra il risultato nell'applicare la trasformata di Hough in Fig. 3.5a.



(a) Risultato dell'applicazione algoritmo di Canny in Fig. 3.3

(b) Risultato dell'applicazione trasformata di Hough in Fig. 3.5a

Figura 3.5

A questo punto si possono estrarre picchi di Hough e linee desiderate.

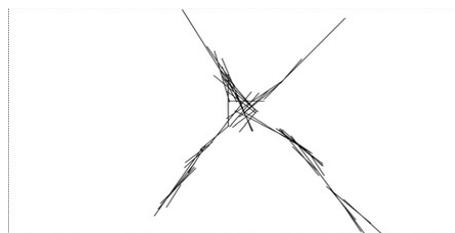


Figura 3.6: Risultato di rilevamento delle rette di Hough basato sui risultati mostrati nelle Figg. 3.5a e 3.5b

Ora entra in gioco il raggruppamento K-means che è stato selezionato per il metodo di partizione. Si suddividono i dati in k gruppi che si escludono a vicenda e si restituisce l'indice del gruppo a cui si è assegnata ogni osservazione. I raggruppamenti di sinistra rappresentano la parete destra e i raggruppamenti di destra rappresentano la parete sinistra e il divario tra loro rappresenta lo spazio disponibile per la navigazione. È stato presto riconosciuto che se il robot si trova al centro del corridoio, la densità di linee non verticale e non orizzontale è quasi lo stesso sui lati destro e sinistro dell'immagine. Se riuscissimo a modellare le zone con più densità di linee, la forma del corridoio comprese le pareti e il divario tra loro sarebbero facilmente distinguibili nelle immagini.

Abbiamo trovato che se si scelgono solo i pixel iniziale e finale di ogni segmento, invece di importare tutti i pixel che si trovano tra i due punti, siamo in grado di diminuire il tempo dell'algoritmo K-means di quasi il 10%. Chiameremo il pixel all'inizio di ogni segmento 'punto iniziale' e il pixel alla fine di esso 'punto finale'. Abbiamo trovato quattro gruppi per il set di dati che comprendono sia punti iniziali che punti finali (Fig. 3.7).



Figura 3.7: Risultato del ritrovamento di quattro gruppi; i cerchi marcatori sono utilizzati per indicare il baricentro di ogni gruppo

Come mostrato in Fig. 3.8, il baricentro di ogni gruppo viene utilizzato per formare un nuovo insieme di dati per un raggruppamento finale. Consideriamo il baricentro finale come il punto di fuga perché la sua posizione è esattamente la stessa della posizione del punto di fuga nell'immagine.

Trovando il punto di fuga possiamo correggere la direzione del robot all'interno



Figura 3.8: Raggruppamento finale e indicazione del punto di fuga

del corridoio ma per fare ciò abbiamo bisogno che ci sia almeno un gruppo per ogni parete. La Fig. 3.9 mostra il motivo per la scelta di quattro gruppi per la classificazione di una serie di dati che comprende i punti di inizio e i punti di fine. Infatti se classifichiamo il set di dati in due gruppi (uno per ogni parete), il risultato finale di raggruppamento su questi baricentri sarà come Figg. 3.9 (a) e (b). Il baricentro finale (segnato da un cerchio) in Fig. 3.9 (a) rappresenta il punto di fuga ma lo stesso baricentro in Fig. (b) non ha le caratteristiche di un punto di fuga poiché non è il punto in cui le linee parallele fra loro ma non parallele al piano dell'immagine, sembrano convergere. Poi è stata fatta la stessa cosa con quattro gruppi o due gruppi per ogni parete. I risultati sono mostrati nelle Fig. 3.9 (c) e (d). Il baricentro risultato è il punto di fuga in entrambi i corridoi. Questo è il motivo per cui sono stati trovati quattro gruppi nell'algoritmo. Questa procedura è stata testata in diversi corridoi ed i risultati sono stati così forti e precisi come in Fig. 3.9 (c) e (d).

Può succedere in alcuni casi che la stima del punto di fuga sia errata a causa di occlusione o in presenza di molti oggetti ingombranti. Quando il robot si muove in linea retta all'interno del corridoio parallelamente alle pareti, la coordinata x del punto di fuga all'interno dell'immagine è vicina a quella del centro dell'immagine C_x . Se la direzione del robot cambia, allora anche la coordinata x del punto di fuga cambia. Il robot utilizza poi un semplice regolatore per mantenere la direzione dritta. Non appena il robot devia dalla sua direzione di linea dritta, il regolatore varia opportunamente le velocità delle due ruote per riportare il robot



Figura 3.9: Classificazione dei dati con due e quattro gruppi

alla sua direzione richiesta. Il robot può muoversi fino a che è vicino a un ostacolo o a persone, cioè fino a che l'algoritmo non riesce più a trovare le linee corrette. Si fermerà in questa situazione finché la condizione normale non sarà di nuovo disponibile. La Fig. 3.14 mostra i risultati relativi.

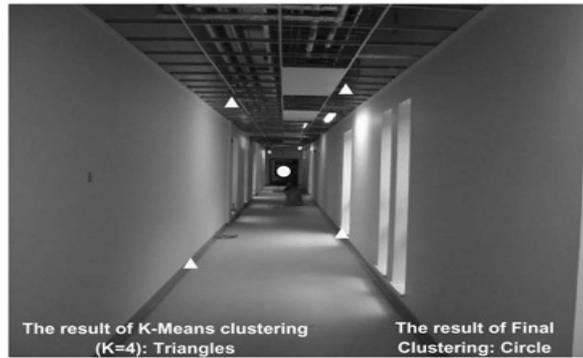
Il metodo proposto è stato applicato in immagini reali e i risultati ottenuti sono molto robusti come nelle successive Fig. 3.10-3.14 in ambienti differenti.



K-Means clustering of data set (K=4)



Final Clustering



The result of K-Means clustering (K=4): Triangles

The result of Final Clustering: Circle

The result of final clustering is our vanishing point

Figura 3.10: Risultati sperimentali 1

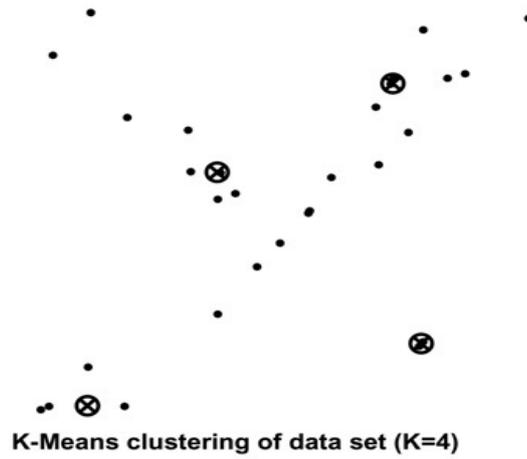


Figura 3.11: Risultati sperimentali 2



Figura 3.12: Risultati sperimentali 3



Figura 3.13: Risultati sperimentali 4



Figura 3.14: Risultati sperimentali 5

Sono stati fatti alcuni tentativi per testare l'algoritmo anche in ambienti esterni. In questo caso sono necessari solo due gruppi per trovare il punto di fuga come è mostrato in Fig. 3.15.

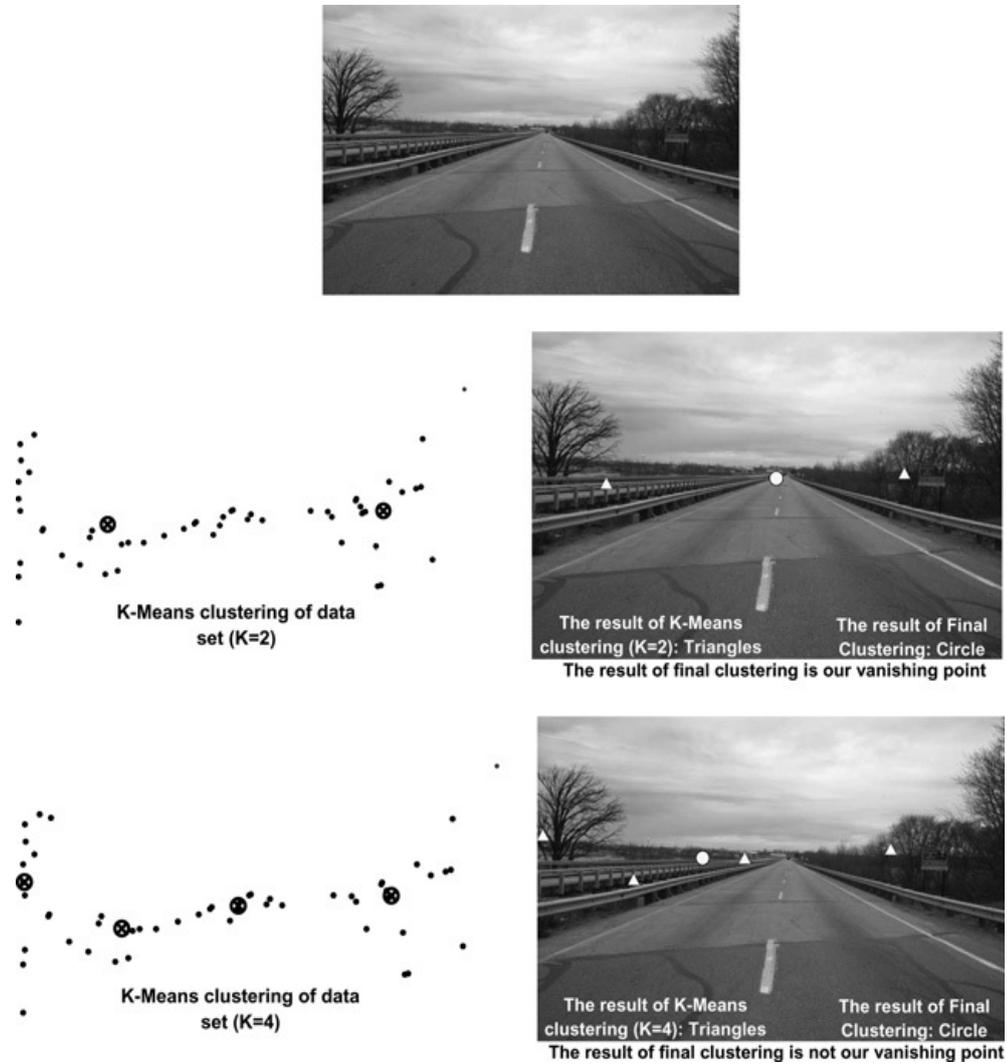


Figura 3.15: Risultati sperimentali in navigazione esterna

Per quanto riguarda la navigazione esterna vedremo alcune osservazioni nel paragrafo seguente.

3.2 Navigazione esterna

Negli ultimi 20 anni la guida autonoma di un veicolo è stato il centro dell'attenzione per molti ricercatori di computer vision. Come si è visto precedentemente un certo sviluppo ha interessato ambienti chiusi [5, 12] ma ha anche raggiunto la navigazione su strada. Queste evoluzioni sono state più lente a causa di problemi di velocità, di sicurezza e di complessità in tempo reale della situazione su strada, tuttavia possediamo oggi significativi progressi. Negli ambienti esterni su strade possiamo distinguere in particolare due situazioni: ambienti strutturati e non strutturati.

In generale, la navigazione all'aperto in un ambiente strutturato richiede una sorta di "road-following" [4]. "Road-following" significa la capacità di riconoscere le linee che separano le corsie o la strada dal muro di sabbia, la struttura della superficie stradale, e le superfici adiacenti, ecc. In sistemi che effettuano la "road-following", i modelli dell'ambiente sono di solito semplici, contenenti solo informazioni come punti di fuga, larghezze stradali e corsie, ecc. "Road-following" per i robot esterni può essere come un "corridoio-following" per robot interni, esclusi i problemi causati da ombre, cambiamento delle condizioni di illuminazione, colori cangianti, ecc. Uno degli approcci più importanti del lavoro nella navigazione esterna è il progetto Navlab e le sue varie trasformazioni.

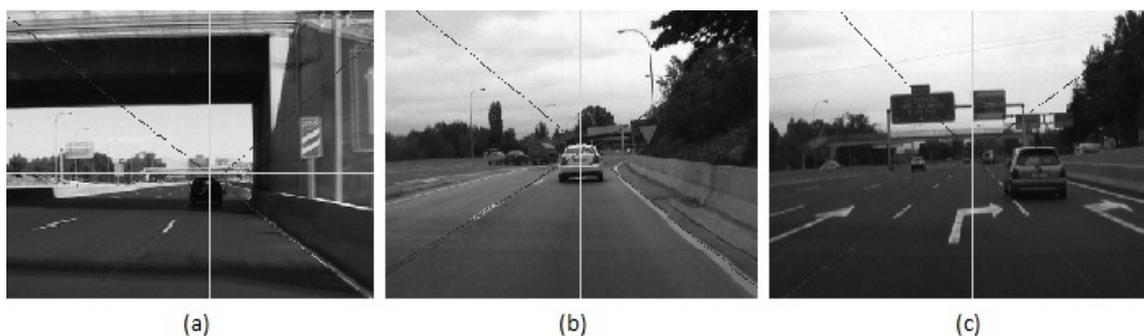


Figura 3.16: Diversi esempi di stima del punto di fuga e di estrazione del modello su strada

Inoltre, l'individuazione automatica della segnaletica stradale (comprese le frecce) e il riconoscimento del testo possono essere molto utili, se non essenziali in alcuni casi, per un veicolo autonomo o anche come aiuto alla consapevolezza della situazione del guidatore in un ambiente stradale sempre più complesso. Sono stati proposti, infatti, dei canali di elaborazione multi-step per il riconoscimento robusto della segnaletica orizzontale e del testo. È dimostrato che questi approcci operano in tempo reale sotto una varietà di condizioni stradali, di guida e di illuminazione [7].

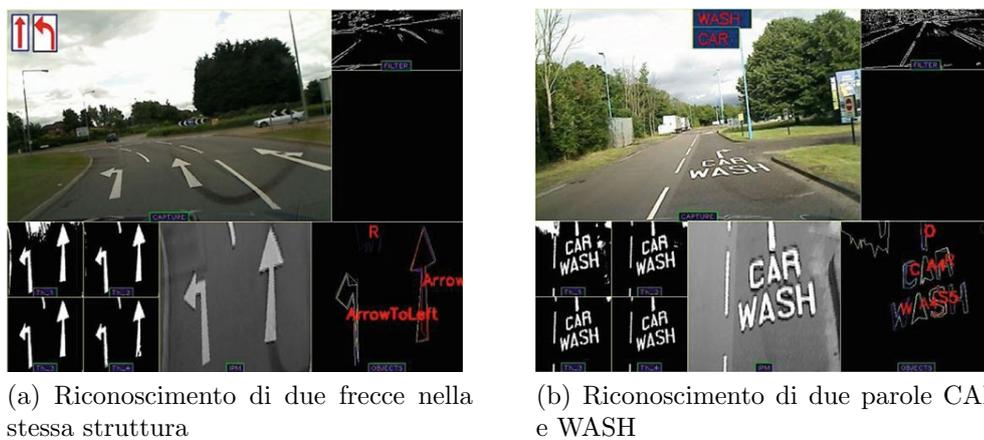


Figura 3.17: Esempi di riconoscimento di segnaletica su strada

Nel caso di ambienti non strutturati ci si imbatte in problematiche che riguardano il rilevamento sulle strade basato sulla visione, perché sono pochissime le caratteristiche distinguibili e invarianti che possono caratterizzare la strada o i confini di tali ambienti. Tuttavia, una caratteristica saliente e coerente della maggior parte delle strade o piste a prescindere dal tipo degli ambienti è che i loro bordi, confini e persino solchi e tracce lasciate dai veicoli precedenti sul percorso sembrano convergere in un singolo punto noto come il punto di fuga. Quindi, la stima di questo punto di fuga gioca un ruolo fondamentale nella determinazione della direzione della strada.

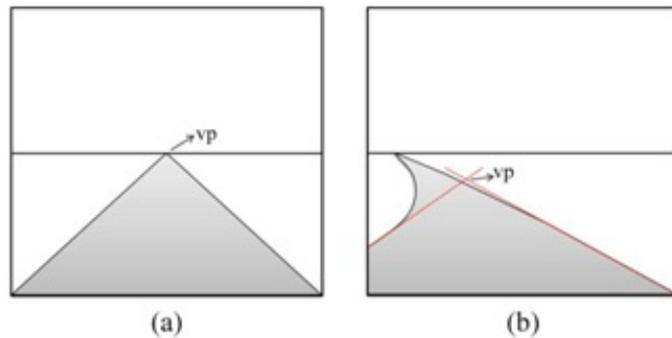


Figura 3.18: Punto di fuga nelle strade (a) dritte e (b) con curve

La maggior parte dei sistemi correnti di rilevamento stradale usa sensori visivi per il rilevamento [19]. I metodi maggiori sono raggruppati in tre categorie principali: basati sul bordo, sulla regione e sulla struttura.

Quelli basati sul bordo riducono la rilevazione della strada all'estrazione dei confini stradali o alla segnaletica orizzontale. Questi approcci sono più appropriati per le strade strutturate le cui linee di demarcazione ben dipinte o i forti limiti di bordo sono le caratteristiche distinte della strada, mentre essi non possono funzionare bene in ambienti non strutturati.

I metodi basati sulla regione cercano caratteristiche distinte, come colore o trama, su intorni locali in modo da suddividere la scena in regioni di percorso e non percorso. Questi metodi sono adatti per i tipi di ambienti in cui vi è almeno un'unica caratteristica di superficie stradale che sia distinguibile dai dintorni. Essi inoltre assumono che le superfici stradali appartengono a regioni relativamente omogenee come in strade ben asfaltate. Tuttavia, in ambienti non strutturati, esistono vari tipi di coperture di terreno (suolo, erba, rocce, ecc) con caratteristiche geometriche complesse, l'illuminazione non controllata, e le condizioni meteorologiche, che rendono distinguibili le regioni della strada da un ambiente estremamente difficile [20].

Per superare le carenze di metodi di un rilevamento corrente di strada, sono state proposte tecniche basate sulla struttura. Invece di cercare localmente spunti



Figura 3.19

distintivi di strada, alcuni autori hanno cercato un vincolo stradale globale per distinguere la direzione della strada [10, 14, 16]. I metodi basati sulla struttura ricercano strutture locali orientate e poi le fanno votare per determinare le posizioni di punti di fuga della strada. Una posizione con il massimo dei voti è considerata come il punto di fuga principale della regione di strada. Inoltre, la direzione della strada o i confini di strada possono essere estratti dalle informazioni della posizione del punto di fuga [13].



Figura 3.20: Il candidato del punto di fuga VP sta ricevendo supporto da possibili elettori di orientamenti di struttura

Il punto di fuga è utilizzato anche in ambienti pedonali [18]. Questi rappresentano una sfida diversa rispetto al chiuso perché sono più aperti con molte meno pareti circostanti, cosa che riduce drasticamente l'efficacia dei sensori di prossimità per dirigere il robot. Allo stesso tempo, le strade pedonali sono molto meno regolamentate rispetto a quelle per la guida delle auto, che forniscono confini e indicazioni ben precise. Per facilitare il processo di riconoscimento, la tecnica solita semplifica la forma della strada (vista dal robot) come un triangolo. Queste ipotesi, però, sono spesso violate in casi in cui vi siano segni complessi, ombre, un misto di strade e piazze, o pedoni sulla strada (Fig. 3.21).



Figura 3.21: Robot che esegue navigazione autonoma in un ambiente esterno non vincolato (campus universitario) e tra le persone. Il robot deve risolvere due sotto-problemi: stimare la direzione di strada e evitare ostacoli

Un altro modo per riconoscere la strada è l'utilizzo del punto di fuga (VP) nell'immagine. La maggior parte dei sistemi usa la direzione prevalente di tessiture locali o "edgels" (edge pixels, pixel dei contorni) per votare per il più probabile VP.

In [18] il sistema rappresenta la strada come un insieme di linee estrapolate dai segmenti di contorno rilevati nell'immagine. Queste linee consentono al robot di mantenere la sua direzione centrando il punto di fuga nel suo campo visivo, e di

correggere la sua deriva a lungo termine dalla sua posizione laterale originale. Si integra l'odometria e il sistema di riconoscimento visivo della strada in una mappa locale a griglia che stima sia la posa del robot sia la zona circostante, al fine di generare un itinerario. Una preoccupazione è come tornare al percorso impostato quando il robot deve deviare da esso momentaneamente evitando nuovi ostacoli (pedoni). Questo sistema è in grado di far fronte a varie sfide come a strade con forme complesse, a strutture di superficie, o marcatore, alle ombre e anche a situazioni affollate. Il sistema completo è in grado di guidare autonomamente il robot su più di 5 km percorsi diversi in un campus universitario densamente popolato.

3.3 Attendibilità delle fonti

Durante il lavoro di questa tesi son state riscontrate alcune incongruenze sul contenuto di un articolo preso in considerazione [11]. Questa fonte dal titolo "VPass: Algorithmic Compass using Vanishing Points in Indoor Environments" presenta un metodo per la navigazione autonoma di veicoli in ambienti interni bastato sul VPass, ovvero una bussola algoritmica che utilizza il punto di fuga per ottenere l'informazione sulla direzione del robot. Si tratta di un algoritmo EKF SLAM che stima l'angolo di rotta del robot e si vuole ricercare un angolo (TI), invariante di traslazione tra rette viste come caposaldi, che non dipenda dalla posizione del robot.

Durante l'analisi del testo ci siamo imbattuti nei seguenti problemi:

Primo problema: il primo problema riguarda la relazione tra la definizione del punto di fuga e le coordinate omogenee con cui è scritto. Gli autori definiscono il punto

di fuga come l'intersezione di due rette $r_1 : ax+by+c_1 = 0$ e $r_2 : ax+by+c_2 = 0$ in coordinate omogenee nel seguente modo $VP = \begin{bmatrix} a \\ b \\ c_1 \end{bmatrix} \times \begin{bmatrix} a \\ b \\ c_2 \end{bmatrix} = (c_2 - c_1) \begin{bmatrix} b \\ -a \\ 0 \end{bmatrix}$

e affermano che fissati a e b ovvero fissata la direzione di queste rette parallele esistono infiniti punti di fuga VPs al variare di $(c_2 - c_1)$. Ma questo non è cor-

retto nel senso che le coordinate omogenee di un punto, proprio o improprio che sia, sono sì infinite perché definite a meno di un fattore di proporzionalità, ma rappresentano un'unica direzione e quindi non è corretto affermare che al variare di $(c_2 - c_1)$ esistono infiniti punti di fuga.

Secondo problema: una seconda perplessità riguarda il concetto di non linearità relativo alla posizione del robot. Gli autori sostengono che ψ^r è una funzione non lineare rispetto alla posizione del robot e nel passaggio da ψ a $\psi + \pi$ sullo stesso semipiano affermano che non c'è linearità, ma se si guarda ad esempio la Fig. 3.22 all'interno dello stesso semipiano in cui sta il robot non c'è dipendenza dalle coordinate del robot x_r e y_r e c'è perfetta linearità con l'angolo θ_r .

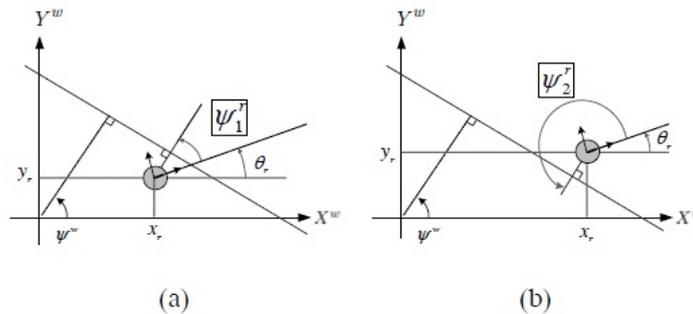


Figura 3.22: Una retta è rappresentata da due diverse angolazioni rispetto alle coordinate del robot

Terzo problema: un terzo problema riguarda la dipendenza dell'angolo ψ^r dalla posizione del robot. Essi definiscono $z = \psi^r = \text{atan2}(dy, dx) - \theta_r$ (*), dove $dx = (\rho^w - x_r \cos \psi^w - y_r \sin \psi^w) \cos \psi^w$ e $dy = (\rho^w - x_r \cos \psi^w - y_r \sin \psi^w) \sin \psi^w$, attraverso cui dimostrano che c'è un'importante dipendenza dalla posizione del robot, ma in realtà questa dipendenza va intesa come in quale semipiano si trova il robot e in altre parole lo si può intendere come una differenza di $+\pi$.

Basterebbe dire che $\psi^r = \begin{cases} \psi^w - \theta_r & \text{se il robot si trova nel semipiano dell'origine} \\ \psi^w + \pi - \theta_r & \text{altrimenti} \end{cases}$.

Quindi la formula (*) era eccessiva e non era necessario rigirla in quella forma per far apparire il tutto non lineare e facendo credere che ψ^r dipendesse in modo

decisivo dalla posizione del robot.

Queste osservazioni riportate vogliono mostrare che non sempre, purtroppo, si ha a che fare con articoli pienamente attendibili, e quindi bisogna sempre verificare la validità delle fonti.

Conclusione

In questo elaborato abbiamo voluto sottolineare l'importante ruolo dei punti di fuga nella robotica mobile. Per cominciare abbiamo analizzato quelli che sono i principali obiettivi degli autori interessati alla navigazione autonoma di robot e siamo andati a selezionarne uno in particolare e ad approfondirlo. Ci siamo interessati al rilevamento della posa del robot all'interno di ambienti interni di un edificio e siamo andati a vedere quali metodi venivano usati dagli autori per risalire alla posizione del robot. L'utilizzo del punto di fuga è uno degli approcci più utilizzati in questo campo e abbiamo selezionato un metodo che ne tenesse conto: abbiamo scelto un approccio basato sulla trasformata di Hough e sul raggruppamento K-means. Abbiamo poi approfondito l'utilizzo dei punti di fuga nella navigazione esterna di robot e in modo molto generale abbiamo affrontato i diversi aspetti sia di ambienti strutturati che non strutturati. In particolare abbiamo affrontato il caso anche di ambienti pedonali, come ad esempio i campus universitari, dove si presenta il problema di navigare evitando ostacoli come possono essere le persone. Infine abbiamo scelto di aggiungere un paragrafo in cui presentare una sottolineatura riguardo l'attendibilità delle fonti, perché durante il lavoro svolto ci siamo imbattuti in un articolo che presentava alcune scorrettezze su certe affermazioni riguardanti l'utilizzo dei punti di fuga e altre nozioni geometriche e abbiamo voluto presentare delucidazioni a riguardo che mettessero in chiaro come non sempre si ha a che fare con fonti pienamente attendibili.

Bibliografia

- [1] R. Bishop. Intelligent vehicle applications worldwide. *Intelligent Systems and their Applications, IEEE*, 15(1):78–81, 2000.
- [2] M. Bosse, R. Rikoski, J. Leonard, and S. Teller. Vanishing points and three-dimensional lines from omni-directional video. *The Visual Computer*, 19(6):417–430, 2003.
- [3] K. Daniilidis and J. Ernst. Active intrinsic calibration using vanishing points. In *Computer Vision and Pattern Recognition, 1996. Proceedings CVPR'96, 1996 IEEE Computer Society Conference on*, pages 708–713. IEEE, 1996.
- [4] G. N. DeSouza and A. C. Kak. Vision for mobile robot navigation: A survey. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(2):237–267, 2002.
- [5] R. Ebrahimpour, R. Rasoolinezhad, Z. Hajiabolhasani, and M. Ebrahimi. Vanishing point detection in corridors: using hough transform and k-means clustering. *Computer Vision, IET*, 6(1):40–51, 2012.
- [6] V. Huttunen and R. Piché. A monocular camera gyroscope. *Gyroscopy and Navigation*, 3(2):124–131, 2012.
- [7] A. Kheyrollahi and T. P. Breckon. Automatic real-time road marking recognition using a feature driven approach. *Machine Vision and Applications*, 23(1):123–133, 2012.

- [8] Y. Kim and S. Kwon. A heuristic obstacle avoidance algorithm using vanishing point and obstacle angle. *Intelligent Service Robotics*, 8(3):175–183, 2015.
- [9] J. Kogecka and W. Zhang. Efficient computation of vanishing points. In *Robotics and Automation, 2002. Proceedings. ICRA '02. IEEE International Conference on*, volume 1, pages 223–228. IEEE, 2002.
- [10] H. Kong, J.-Y. Audibert, and J. Ponce. General road detection from a single image. *Image Processing, IEEE Transactions on*, 19(8):2211–2220, 2010.
- [11] Y. H. Lee, C. Nam, K. Y. Lee, Y. S. Li, S. Y. Yeon, and N. L. Doh. Vpass: Algorithmic compass using vanishing points in indoor environments. In *IROS*, pages 936–941, 2009.
- [12] E. Marder-Eppstein, E. Berger, T. Foote, B. Gerkey, and K. Konolige. The office marathon. 2010.
- [13] P. Moghadam, J. A. Starzyk, and W. S. Wijesoma. Fast vanishing-point detection in unstructured environments. *Image Processing, IEEE Transactions on*, 21(1):425–430, 2012.
- [14] M. Nieto and L. Salgado. Real-time vanishing point estimation in road sequences using adaptive steerable filter banks. In *Advanced Concepts for Intelligent Vision Systems*, pages 840–848. Springer, 2007.
- [15] J. Omedes, G. López-Nicolás, and J. J. Guerrero. Omnidirectional vision for indoor spatial layout recovery. In *Frontiers of Intelligent Autonomous Systems*, pages 95–104. Springer, 2013.
- [16] C. Rasmussen. Grouping dominant orientations for ill-structured road following. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 1, pages I–470. IEEE, 2004.
- [17] R. Schuster, N. Ansari, and A. Bani-Hashemi. Steering a robot with vanishing points. *Robotics and Automation, IEEE Transactions on*, 9(4):491–498, 1993.

-
- [18] C. Siagian, C.-K. Chang, and L. Itti. Mobile robot navigation system in outdoor pedestrian environment using vision-based road recognition. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 564–571. IEEE, 2013.
- [19] C. Thorpe, M. H. Hebert, T. Kanade, and S. A. Shafer. Vision and navigation for the carnegie-mellon navlab. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 10(3):362–373, 1988.
- [20] S. Thrun, M. Montemerlo, H. Dahlkamp, D. Stavens, A. Aron, J. Diebel, P. Fong, J. Gale, M. Halpenny, G. Hoffmann, et al. Stanley: The robot that won the darpa grand challenge. *Journal of field Robotics*, 23(9):661–692, 2006.
- [21] L.-L. Wang and W.-H. Tsai. Camera calibration by vanishing lines for 3-d computer vision. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (4):370–376, 1991.