

UNIVERSITÀ DEGLI STUDI DI PARMA

---

Facoltà di Ingegneria

Corso di Laurea Specialistica in Ingegneria Informatica

REALIZZAZIONE DI UN SISTEMA  
PER LA NAVIGAZIONE EFFICIENTE  
DI LGV IN AMBIENTI QUASI-STATICI

A SYSTEM FOR EFFICIENT LGV NAVIGATION IN  
QUASI-STATIC ENVIRONMENTS

Relatore:

Chiar.mo Prof. STEFANO CASELLI

Correlatore:

Ing. DARIO LODI RIZZINI

Tesi di Laurea di:  
GIONATA BOCCALINI

---

ANNO ACCADEMICO 2010-2011

Ai miei genitori.

*“Se ho visto più lontano, è perchè stavo sulle spalle di giganti.”*

*Isaac Newton*

# Ringraziamenti

Questa tesi è il risultato dell'impegno che ho impiegato nel completare il mio percorso di studi, con tutte le varie difficoltà e soddisfazioni annesse. Naturalmente niente di tutto questo sarebbe stato possibile senza il supporto e l'incoraggiamento della mia famiglia, in particolare dei miei genitori, che ringrazio profondamente per avermi dato la possibilità di proseguire negli studi, e per la pazienza avuta in tutti questi anni!

Desidero anche ringraziare tutte le persone che ho incontrato durante questo periodo, a partire dal Prof. Caselli che mi ha accettato per ben due volte come tesista e mi ha offerto tutta la sua disponibilità e pazienza, l'Ing. Dario Lodi Rizzini per le infinite discussioni sui dettagli, sugli aspetti più nascosti di ogni cosa, sulle nozioni e sulle tecniche che sono alla base di questo e di altri lavori. Il bagaglio culturale, tecnico e non solo, che ho acquisito in questa esperienza mi è stato molto utile e spero di farne buon uso.

Un enorme ringraziamento va a tutti i miei amici vecchi e nuovi, per i momenti piacevoli e divertenti passati con loro. In rigoroso ordine alfabetico anche se qualcuno potrebbe rimanere escluso, e me ne scuso in anticipo, vorrei ringraziare Alice, Andrea, Barbara, Beppe, Carlo, Chiara, Cristian, Dario, Eric, tutti i Fabio, Federico, Filippo, Giovanni, Jose, tutti i Lorenzo, Luca, Maristella, Michele, Mirco, Pietro, Riccardo, Stefano ecc... ecc... ecc...

Grazie a tutti!!

# Indice

<b>Introduzione</b>	<b>1</b>
<b>1 Stato dell'arte</b>	<b>4</b>
1.1 Metodi per la risoluzione di mappe . . . . .	5
1.1.1 Metodi per la localizzazione del robot . . . . .	8
1.2 Tipi di mappe . . . . .	12
1.2.1 Mappe a griglia di occupazione . . . . .	12
1.2.2 Mappe di poligonali . . . . .	14
1.2.3 Mappe di features . . . . .	16
1.3 Costruzione incrementale di mappe locali . . . . .	18
<b>2 Requisiti del problema e scelte progettuali</b>	<b>21</b>
2.1 Descrizione del problema . . . . .	21
2.2 Condizioni operative critiche . . . . .	22
2.3 Riconoscimento di oggetti quasi-statici e dinamici . . . . .	24
2.3.1 Accumulo di scansioni . . . . .	24
2.3.2 Confronto di sottomappe . . . . .	26
2.4 Soluzione per l'individuazione di oggetti quasi-statici . . . . .	29
2.5 Rappresentazione degli oggetti . . . . .	31
<b>3 Algoritmi per la costruzione di mappe locali di poligonali</b>	<b>35</b>
3.1 Estrazione di poligonali . . . . .	36
3.2 Associazione . . . . .	39
3.2.1 Similarità . . . . .	39

---

3.2.2	Procedura di associazione . . . . .	42
3.3	Filtraggio . . . . .	44
3.4	Fusione di poligoni . . . . .	46
3.5	Unione di poligoni . . . . .	50
3.5.1	Riordino delle poligoni . . . . .	51
3.5.2	Algoritmo . . . . .	53
<b>4</b>	<b>Localizzazione e setup sperimentale</b>	<b>56</b>
4.1	Localizzazione . . . . .	59
4.1.1	Triangolazione . . . . .	60
4.1.2	Localizzatore con filtro di Kalman . . . . .	62
4.2	Inseguimento di un percorso pianificato . . . . .	66
4.2.1	Generazione dei controlli . . . . .	67
<b>5</b>	<b>Risultati</b>	<b>72</b>
5.1	Distanza tra poligoni . . . . .	75
5.2	Risultati delle simulazioni . . . . .	76
5.3	Risultati dei test sul robot . . . . .	79
5.3.1	Test in presenza di oggetti dinamici . . . . .	83
<b>6</b>	<b>Modello probabilistico a grafo</b>	<b>86</b>
6.1	SLAM . . . . .	87
6.1.1	Formulazione probabilistica dello SLAM . . . . .	88
6.1.2	Formulazione a grafo dello SLAM . . . . .	89
6.2	Introduzione all'SPmodel . . . . .	92
6.2.1	Corrispondenza tra features . . . . .	93
6.2.2	Rappresentazione dell'incertezza . . . . .	95
6.3	Un framework probabilistico per il problema di SLAM: SPmap	96
6.3.1	Integrazione nel modello a grafo . . . . .	99
6.3.2	Risoluzione del grafo . . . . .	101
	<b>Conclusioni</b>	<b>105</b>

---

<b>A</b>	<b>Configurazione aree del sensore laser</b>	<b>107</b>
<b>B</b>	<b>Jacobiani della funzione di vincolo</b>	<b>111</b>
	<b>Bibliografia</b>	<b>113</b>

# Elenco delle figure

1.1	Loop closure . . . . .	6
1.2	Triangolazione . . . . .	9
1.3	Triangolazione: singolarità . . . . .	9
1.4	Mappa a griglia . . . . .	13
1.5	Mappa a poligonali . . . . .	15
1.6	similarità . . . . .	19
2.1	Aree di sicurezza LGV . . . . .	22
2.2	Confronto scansioni . . . . .	25
2.3	Confronto sottomappe . . . . .	27
2.4	Sottomappe temporanee . . . . .	29
2.5	Scansione ottenuta in un corridoio . . . . .	30
2.6	Poligonale . . . . .	32
2.7	Mappa ottenuta dalla scansione . . . . .	33
3.1	Rimozione outlier . . . . .	37
3.2	Scansione suddivisa in tratti continui . . . . .	37
3.3	Scansione e poligonali estratte . . . . .	38
3.4	Istogramma di una poligonale . . . . .	40
3.5	Funzione tangente di una poligonale . . . . .	41
3.6	Distanza tra due poligonali . . . . .	42
3.7	Associazioni di poligonali . . . . .	43
3.8	Distanza filtraggio poligonali . . . . .	45
3.9	Fusione di due poligonali . . . . .	46

---

3.10	Fusione di una scansione con la mappa . . . . .	47
3.11	Unione di due poligoni . . . . .	51
3.12	Visibilità di una poligonale . . . . .	52
3.13	Distanza unione poligoni . . . . .	53
4.1	Localizzazione LGV . . . . .	57
4.2	Markers disposti nell'ambiente . . . . .	58
4.3	Triangolazione . . . . .	60
4.4	Circonferenze passanti per marker . . . . .	61
4.5	Correzioni del filtro di Kalman . . . . .	65
4.6	Robot Pioneer P3DX . . . . .	68
4.7	Modello robot unicycle . . . . .	69
4.8	Robot di riferimento . . . . .	69
5.1	Ambiente di test: palazzina 1 . . . . .	73
5.2	Distanza tra poligoni, per i test . . . . .	76
5.3	Mappa in simulazione . . . . .	77
5.4	Mappa in simulazione . . . . .	78
5.5	Mappa palazzina 1 . . . . .	79
5.6	Ambiente di test: palazzina 3 . . . . .	80
5.7	Mappa palazzina 3a . . . . .	81
5.8	Mappa palazzina 3b . . . . .	82
5.9	Oggetti dinamici 1 . . . . .	84
5.10	Oggetti dinamici 1 . . . . .	85
6.1	Dynamic Bayesian Network . . . . .	89
6.2	graphSlam . . . . .	90
6.3	Vincolo osservazione predetta . . . . .	91
6.4	Simmetrie di una feature . . . . .	93
6.5	Simmetrie di un vertice . . . . .	93
6.6	Matrici di selezione punto retta . . . . .	95
6.7	SPmap . . . . .	98
6.8	Grafo di supporto per SPmap . . . . .	101

---

6.9	Grafo corretto . . . . .	104
A.1	warning area . . . . .	108
A.2	protective area . . . . .	109
A.3	formato pacchetto dati ethernet del laser LMS111 . . . . .	109

# Introduzione

La guida automatica di veicoli rappresenta un problema molto discusso e studiato dalle comunità di ricerca nel campo della robotica mobile, e recentemente anche da alcune aziende. I principali requisiti richiesti in questo contesto sono rappresentati dall'autonomia del veicolo e dalla sicurezza delle persone e delle cose che risiedono nell'ambiente di lavoro, e possono essere raggiunti raccogliendo informazioni accurate sull'ambiente e sviluppando procedure e politiche di gestione del veicolo e delle situazioni straordinarie che possono accadere. Queste informazioni possono riguardare gli ostacoli presenti e la posizione attuale del robot, e possono essere organizzate e catalogate per facilitare la fase decisionale che il veicolo deve svolgere autonomamente. Le applicazioni in cui si fa uso di procedure di guida automatica si diversificano in base alle situazioni e ai tipi di veicoli utilizzati: per esempio nel campo dell'*automotive* sono stati sviluppati sistemi di assistenza alla guida o sistemi di parcheggio automatico, mentre in altri contesti più controllati le tecniche utilizzate permettono di compiere azioni più articolate. È questo il caso della movimentazione di merci in un magazzino, dove più *Veicoli a Guida Laser* (LGV) vengono gestiti e coordinati per completare lo stoccaggio dei materiali nel più breve tempo possibile, mantenendo elevati standard di sicurezza per le persone e le cose presenti.

Per raggiungere questi requisiti fondamentali gli LGV devono possedere dei sensori per il riconoscimento degli ostacoli presenti nell'ambiente, e delle procedure di rallentamento e fermata di emergenza nel caso di pericolo immediato. Proprio queste situazioni di pericolo influiscono negativamente sulle

prestazioni dell'LGV. Infatti nel caso in cui le rilevazioni sensoriali indichino la presenza di ostacoli nelle immediate vicinanze, il veicolo, per motivi di sicurezza, deve rallentare oppure fermarsi, con conseguente riduzione dell'efficienza operativa rispetto a quella pianificata in condizioni ottimali.

Tra i possibili tipi di ostacoli che si possono individuare in un ambiente industriale particolare importanza viene acquisita dagli ostacoli definiti come quasi-statici: sono oggetti che determinano percezioni sensoriali nelle aree di navigazione e di sicurezza degli LGV e ne possono provocare il rallentamento, nelle zone in cui vengono rilevati, anche per tempi prolungati. Tuttavia questi oggetti non rappresentano l'infrastruttura statica dell'ambiente, e quindi la loro posizione e dimensioni non può essere nota a priori, anche disponendo di una mappa statica dei locali in cui il veicolo lavora.

Lo scopo di questa tesi è sviluppare un metodo affidabile per identificare gli oggetti quasi-statici, utilizzando solo le informazioni sensoriali provenienti dai sensori montati sulle navette, in modo da consentire agli sviluppatori del sistema di controllo e navigazione degli LGV di adottare le politiche di gestione che ritengono più appropriate, finalizzate al miglioramento delle prestazioni del sistema di gestione degli LGV anche in presenza di ostacoli quasi-statici o dinamici.

Sono state considerate alcune soluzioni mirate ad ottenere una rappresentazione coerente dell'ambiente. Data la strutturazione del movimento degli LGV, suddiviso in brevi segmenti di traiettoria, la soluzione ritenuta più conforme a questo contesto applicativo consiste nell'impiego di mappe locali adatte a rappresentare porzioni limitate dello spazio di lavoro. Considerando la natura dei sensori laser a disposizione e la necessità di una rappresentazione compatta è stato deciso di utilizzare mappe composte da archi poligonali, che garantiscono una buona precisione nella ricostruzione dell'ambiente e possono essere facilmente elaborate e condivise.

La tesi è organizzata come segue. Il capitolo 1 fornisce una panoramica delle tecniche e rappresentazioni descritte in letteratura per la mappatura dell'ambiente, per le elaborazioni necessarie al mantenimento della mappa durante

il movimento del veicolo, per l'identificazione di ostacoli quasi-statici e dinamici, e per la localizzazione del veicolo rispetto ad un sistema di riferimento globale. La prima parte del lavoro è concentrata sulle scelte di una rappresentazione adatta per il dominio applicativo, considerando i requisiti e gli scopi finali del progetto. Le motivazioni e i dettagli relativi a questa scelta vengono descritti nel capitolo 3. Successivamente vengono sviluppati alcuni algoritmi per ottenere mappe locali consistenti utilizzando la rappresentazione scelta, partendo da sequenze di scansioni consecutive, e mantenendo le mappe aggiornate, anch'essi descritti nel capitolo 3. Il capitolo 4 descrive i modelli e le procedure algoritmiche utilizzate al fine di testare sperimentalmente il funzionamento del sistema proposto, per i quali è stato necessario ricostruire lo scenario in cui operano gli LGV sulla piattaforma robotica disponibile in laboratorio. Successivamente sono stati eseguiti alcuni test sia in simulazione che in ambienti reali; tali test riguardano la creazione della mappa dell'ambiente e i successivi passaggi del robot che permettono di stimare la precisione del riconoscimento di oggetti già osservati, anche in presenza di oggetti dinamici nell'area di scansione. I risultati ottenuti sono presentati nel capitolo 5.

Infine, il capitolo 6, illustra un modello probabilistico a grafo utilizzabile per irrobustire la rappresentazione dell'ambiente scelta ed essere in grado di creare mappe globali di grandi dimensioni, mantenendo la precisione nella ricostruzione dell'ambiente.

# Capitolo 1

## Stato dell'arte

Questo lavoro di tesi ha come obiettivo principale il riconoscimento di oggetti tramite l'uso di un robot mobile equipaggiato con sensori di prossimità, e la classificazione di questi oggetti in statici, quasi-statici e dinamici. Di conseguenza il robot deve costruire la mappa (o una parte della mappa) dell'ambiente, identificando al contempo le posizioni degli oggetti riconosciuti. Per fare questo il robot deve conoscere la propria posizione, e quindi attuare, almeno localmente, una procedura di localizzazione. Le fasi di apprendimento della mappa e localizzazione al suo interno rappresentano un problema generale molto discusso nella letteratura inerente alla robotica mobile, e noto come *Simultaneous Localization And Mapping* (SLAM) [17]. La costruzione della mappa implica l'integrazione continua di dati provenienti dai sensori del robot nella rappresentazione scelta. La localizzazione consiste nel stimare la posa attuale del robot relativamente alla mappa ottenuta, sia quando la posa iniziale è nota, sia quando non si conosce a priori la configurazione iniziale del robot. I due sottoproblemi non possono essere risolti indipendentemente, dal momento che la costruzione di una buona mappa è un requisito fondamentale per la fase di localizzazione, mentre la stima della posizione attuale del robot è necessaria per posizionare correttamente gli elementi della mappa.

In questo capitolo vengono introdotti i lavori presenti in letteratura che sono stati considerati nella soluzione proposta in questa tesi: nella sezione 1.1

vengono descritte le motivazioni e le problematiche inerenti al problema dello SLAM, insieme ad alcuni metodi di risoluzione di mappe e localizzazione del robot. Nella sezione 1.2 vengono descritti i diversi tipi di mappe e rappresentazioni esistenti, Nella sezione 1.3 vengono descritti i metodi di integrazione di dati sensoriali per la creazione di mappe e finalizzati al riconoscimento della natura statica o dinamica degli oggetti.

## 1.1 Metodi per la risoluzione di mappe

Molti dei compiti che un robot mobile può eseguire necessitano di una mappa per essere completati in modo efficiente. La disponibilità della mappa consente al robot di navigare basandosi solo sull'utilizzo dei sensori con cui è equipaggiato, senza richiedere la presenza di altri tipi di sensori esterni o di particolari condizioni dell'ambiente operativo. La creazione di una mappa richiede l'accumulo e l'integrazione dei dati sensoriali acquisiti dal robot durante l'esplorazione dell'ambiente. Una mappa costruita tramite la sola odometria del robot è affetta dall'incertezza associata a questo tipo di sensore. Nel caso della costruzione di una mappa locale, date le piccole dimensioni, questo problema potrebbe essere trascurabile, e la mappa potrebbe mantenere una struttura consistente. Quando invece la mappa creata assume dimensioni maggiori l'errore accumulato dall'odometria del robot durante il movimento non può esser trascurato e la precisione nella costruzione della mappa può essere gravemente compromessa. Per questi motivi è indispensabile, soprattutto nel caso di mappe globali su larga scala, impiegare dei metodi di risoluzione della mappa per ottenere la configurazione dei suoi elementi che meglio rappresenta lo spazio esplorato.

Uno dei principali problemi riscontrabili nell'ambito dello SLAM è chiamato *data association* [46] [47], e consiste nella associazione degli elementi estratti dall'osservazione corrente con alcuni degli elementi della mappa che rappresentano la stessa zona dell'ambiente. Infatti quando il robot visita una porzione dello spazio che era già stata visitata in precedenza deve essere in

grado di riconoscere la situazione e associare le osservazioni con quelle già presenti nella mappa. La corretta interpretazione di questo problema, noto anche come *loop closure* a causa del ciclo di visita del robot (figura 1.1), consente di ottenere informazioni aggiuntive sulle relazioni tra gli elementi della mappa, che sono indispensabili per la riuscita della procedura di risoluzione. I metodi di risoluzione possono essere suddivisi in due categorie principali:

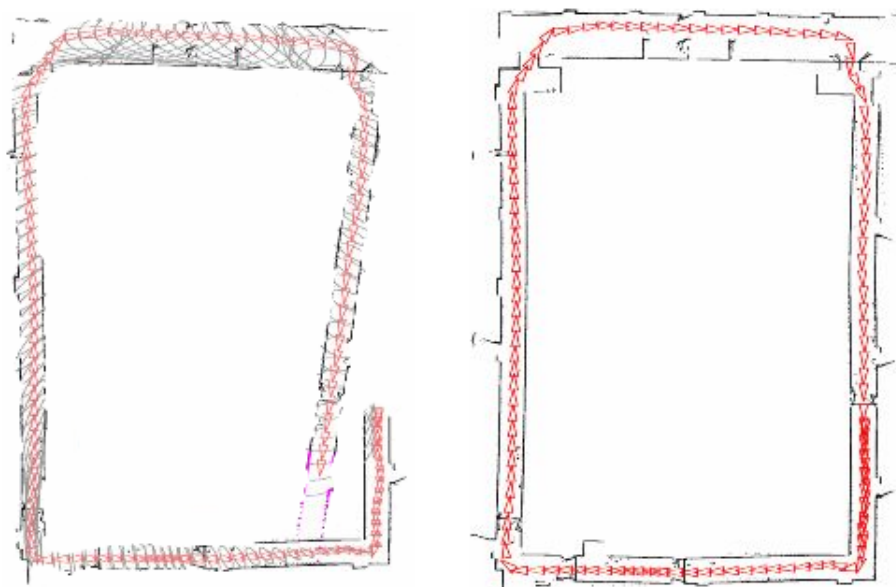


Figura 1.1: Mappe ottenute dalla navigazione di un robot in un ambiente circolare: a sinistra la mappa creata prima della chiusura del loop, mentre a destra l'operazione di associazione consente di correggere la mappa e ottenere una rappresentazione consistente.

- metodi di filtraggio: in questo caso il problema della risoluzione della mappa viene interpretato come una stima dello stato di un filtro Bayesiano, dove lo stato consiste nella posa corrente del robot e nella mappa stessa. La stima viene migliorata durante il movimento del robot incorporando nuovi dati sensoriali non appena sono disponibili. In questa categoria ricadono alcune tecniche basate su filtri di Kalman [27], filtri particellari [18], o su filtri informazione [19]. Questi approcci, come

conseguenza della loro natura incrementale, possono essere eseguiti in linea nella maggior parte dei casi;

- metodi di stima completa: questi metodi prevedono di stimare la traiettoria completa del robot, partendo dalla serie completa di osservazioni [20], [21]. Questi approcci affrontano il cosiddetto problema del *full SLAM*, e in genere si basano su tecniche di minimizzazione dell'errore quadratico generato dall'intero insieme di dati.

Un'ulteriore classificazione dei metodi di risoluzione è essere basata sulla natura probabilistica del metodo stesso. I metodi probabilistici fanno uso della teoria della probabilità per stimare la distribuzione di probabilità associata alla traiettoria del robot e alla posizione degli elementi nella mappa. Frequentemente l'uso di questi metodi viene coniugato con una particolare struttura a grafo per la mappa, che viene costruita partendo dai dati sensoriali. I nodi del grafo rappresentano le pose del robot o i landmark osservati, mentre gli archi modellano le relazioni spaziali tra i nodi. Un arco che congiunge due pose descrive lo spostamento relativo che il robot ha compiuto, mentre un arco che congiunge una posa con un landmark rappresenta l'osservazione effettuata. Ad ogni arco è associato un vincolo che consiste in una distribuzione di probabilità sulla trasformazione relativa tra i due nodi connessi. Questa formulazione del problema dello SLAM è denominata Graph-SLAM, ed è stata introdotta in [20]. In questa tesi è stato utilizzato un modello probabilistico a grafo, descritto nel capitolo 6, per estendere le funzionalità dell'applicazione anche al caso di costruzione di mappe globali di grandi dimensioni.

Una volta che il grafo è stato costruito è necessario trovare la configurazione di nodi che meglio si adatta alle misurazioni sensoriali modellate dagli archi del grafo, cioè quella configurazione che minimizza l'errore complessivo sulle osservazioni e sulla posa del robot. Questo implica la risoluzione della rete di vincoli, per la quale alcuni metodi applicano una tecnica di rilassamento dei vincoli per ottenere la posa del robot e costruire la mappa [22]. Altri ampliano questa tecnica e la utilizzano per ottimizzare la mappa a diversi

livelli di risoluzione [23]. Altri ancora cercano di ridurre la complessità del problema con tecniche di ottimizzazione non lineare anche in presenza di grandi errori iniziali sull'odometria [24] o tramite eliminazione di variabili [25] per diminuire i tempi necessari alla risoluzione, anche con grafi di gradi dimensioni.

### 1.1.1 Metodi per la localizzazione del robot

Una fase critica del problema di riconoscimento di oggetti quasi-statici e dinamici è rappresentata dalla procedura di localizzazione, che permette al robot di individuare le corrette posizioni degli oggetti e di inseguire una traiettoria pianificata a priori, compito necessario per le prove simulate effettuate per testare il sistema. In questo lavoro di tesi la localizzazione è basata sull'utilizzo di marker artificiali, e consiste in una procedura di triangolazione che è in grado di ricavare la posa assoluta del robot note le posizioni dei marker nell'ambiente, e le rispettive osservazioni acquisite tramite il sensore laser. La posa ottenuta da questa procedura permette di inizializzare il vettore di stato di un filtro di Kalman esteso impiegato nel tracciamento della posa del robot, come descritto nel capitolo 4.

In [38] viene descritto un metodo di triangolazione geometrico denominato *Generalized Geometric Triangulation* che fa uso di almeno tre markers per individuare univocamente la posizione del robot, e per questo rientra nella categoria di algoritmi di triangolazione chiamata *Three Object Triangulation* [39]. Nel caso siano visibili più di tre marker solo i tre più vicini al robot vengono considerati. Di conseguenza tutte le aree dell'ambiente in cui non sono visibili almeno tre marker non possono essere utilizzate per la localizzazione. Note le posizioni dei markers ricavate tramite le osservazioni sensoriali l'algoritmo calcola due circonferenze, formate dai marker  $L_1$  e  $L_2$ , e dai marker  $L_1$  e  $L_3$ , come mostrato in figura 1.2. Il punto di intersezione tra le due circonferenze rappresenta la posizione del robot e tramite un ragionamento geometrico l'algoritmo è in grado di identificare anche l'orientazione del robot, completando la posa globale. Quando tutti i marker e il robot si trovano

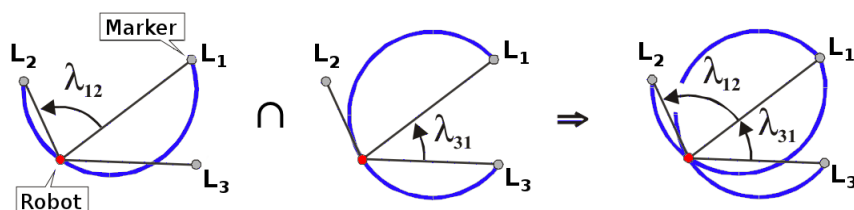


Figura 1.2: Circonferenze passanti per le due coppie di marker e loro intersezione per ottenere la posa del robot mediante triangolazione.

sulla stessa circonferenza il problema della triangolazione non ha una unica soluzione (infatti l'intersezione delle due circonferenze è ancora una circonferenza e non un punto, come mostrato in figura 1.3). Quindi in questa zona la localizzazione non può avvenire, e nello spazio adiacente alla circonferenza l'errore di localizzazione alto non permette la corretta identificazione della posizione del robot.

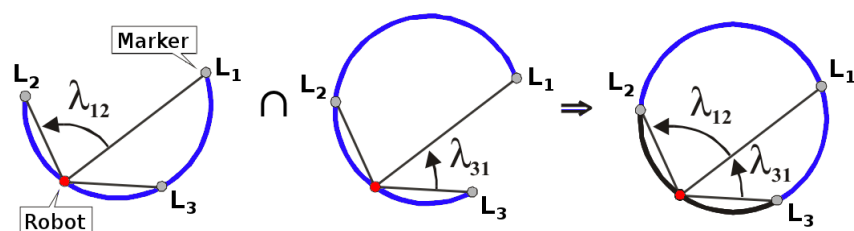


Figura 1.3: I marker e il robot si trovano sulla stessa circonferenza, quindi l'intersezione è ancora una circonferenza, e non si può ottenere un unico punto come soluzione della triangolazione.

Tuttavia questo algoritmo di triangolazione non risente di alcune limitazioni presenti invece nell'algoritmo originale descritto in [39], che sono:

- i markers devono essere disposti e etichettati in senso antiorario;
- gli angoli  $\lambda_{12}$  e  $\lambda_{31}$  (in figura 1.2) devono essere minori di  $180^\circ$ , altrimenti le etichette devono essere scambiate fino al soddisfacimento di questa condizione;
- l'algoritmo è affidabile solo quando il robot si trova all'interno del triangolo formato dai tre markers.

Quindi i vantaggi dell'algoritmo presentato si possono riassumere nei seguenti punti:

- etichettatura dei marker in qualunque ordine;
- possibilità di disporre i marker in qualunque posizione;
- gli angoli  $\lambda_{ij}$  possono avere anche valori maggiori di  $180^\circ$ ;
- affidabilità dell'algoritmo anche all'esterno o in corrispondenza del triangolo formato dai markers.

Il metodo descritto in [40] ricade nella stessa categoria del precedente, ma cerca di migliorare la precisione della fase di triangolazione geometrica considerando noti la mappa dell'ambiente e le posizioni dei marker. Infatti il problema principale nell'applicazione di una tecnica di triangolazione è rappresentato dalle misurazioni effettuate dal sensore che, durante il movimento del robot, possono essere acquisite da pose diverse in diversi istanti di tempo; questo impedisce di applicare direttamente e in modo consistente la triangolazione con le misurazioni ottenute. Il metodo in questione propone l'utilizzo di un filtro di Kalman esteso per risolvere questo problema, il cui vettore di stato è composto dagli angoli misurati tra il robot e i tre marker osservati. Il filtro tiene traccia dell'orientazione del robot tramite l'odometria e ottiene la stima ottima degli angoli contenuti nel vettore di stato. Quindi gli angoli stimati possono essere utilizzati per applicare la procedura di triangolazione. In più il modello delle osservazioni contenuto nel filtro è lineare, dal momento che i suoi componenti sono misurati direttamente dal sensore, e questo migliora la convergenza del filtro rispetto all'utilizzo di un modello non lineare. Il metodo proposto può essere applicato con un qualunque numero di marker, ma risente comunque del problema descritto per il metodo precedente, comune a tutti gli algoritmi di triangolazione: il caso in cui il robot si trova sulla stessa circonferenza passante per i marker rappresenta una singolarità del problema e impedisce di ottenere la posa.

In [41] viene presentato un metodo per la localizzazione globale di veicoli LGV in contesti industriali, dove la mappa dell'ambiente è nota a priori così come la posizione dei markers riflettenti. Il problema consiste nell'identificare la posizione globale dell'LGV senza alcuna stima a priori, sia al momento dell'accensione del veicolo sia quando il veicolo perde le informazioni riguardo alla propria posizione e quindi deve essere reinizializzato. Questo problema può essere suddiviso in due fasi: nella prima fase i marker osservati nella scansione corrente devono essere associati con quelli definiti nella mappa, cioè deve essere trovata una corrispondenza uno a uno tra i due insiemi di markers. In una applicazione industriale questo compito può essere complicato dalla natura dinamica degli oggetti presenti nell'ambiente e dalla presenza di materiali con proprietà di riflettività simile ai marker utilizzati, che possono provocare falsi positivi. Per risolvere in parte questi problemi vengono utilizzati marker dalle proprietà e forme particolari, che sono disposti in modo capillare nell'ambiente, anche in grande numero. Oltre a questo il metodo è in grado di filtrare le osservazioni errate e riconoscere i falsi positivi o falsi negativi nell'identificazione dei markers, utilizzando una singola scansione laser, senza alcuna procedura di fusione sensoriale, per l'intero algoritmo. La seconda fase consiste nella determinazione della posa dell'LGV rispetto a un sistema di riferimento globale, tramite il confronto delle posizioni relative e assolute dei marker individuati e di quelli presenti nella mappa.

Tramite queste due fasi il problema della localizzazione automatica (senza l'intervento di un operatore) di un LGV in un ambiente noto viene risolto con una buona precisione e in tempi brevi, e quindi consente di aumentare l'autonomia di tutto il sistema composto da più LGV che cooperano e condividono informazioni sulle loro posizioni e sui compiti che stanno eseguendo, e di recuperare il normale funzionamento anche dopo un problema nella localizzazione di un veicolo.

## 1.2 Tipi di mappe

Nell'ambito della robotica mobile esistono diversi tipi di mappe che differiscono per tipo di rappresentazione adottata, livello di dettaglio ottenibile, algoritmi di costruzione e aggiornamento, occupazione di memoria, scalabilità, facilità di gestione, ecc.. Di seguito vengono riportati i principali tipi di mappe presenti in letteratura.

### 1.2.1 Mappe a griglia di occupazione

Nelle mappe a griglia l'ambiente è rappresentato da una matrice di celle, dove ogni cella contiene informazioni riguardo alla probabilità che il corrispondente spazio si occupato [1]. In alcuni casi le celle possono essere etichettate in base a tre valori, libera, occupata o sconosciuta, oppure possono essere associate a dei valori di densità di probabilità in un intervallo continuo. La mappa è costruita proiettando le misurazioni ottenute dal sensore sulla matrice di celle discrete, e combinando queste informazioni con i valori provenienti dalle altre misurazioni e dal valore contenuto nelle celle allo stato precedente. Infatti più misurazioni nella scansione corrente per la stessa cella permettono di rinforzare i valori di probabilità contenuti nella cella stessa, se l'osservazione è concorde (una cella libera viene confermata come libera) oppure modificare il valore di probabilità se l'osservazione è discorde (una cella precedentemente libera viene osservata come occupata, o viceversa). Durante l'aggiornamento le distribuzioni di probabilità sulle celle libere e su quelle occupate non hanno lo stesso significato, quindi le operazioni svolte per l'integrazione di più osservazioni sono diverse a seconda della cella che si sta esaminando. Combinando più scansioni dell'ambiente mentre il robot si muove è possibile ottenere mappe sempre più precise, in cui il livello di incertezza riguardo alle zone libere e occupate dell'ambiente diminuisce. In più, tramite algoritmi di normalizzazione e soglia, si ottengono mappe con una buona precisione e con un alto contenuto informativo. Mappe di questo tipo possono essere costruite utilizzando sensori di prossimità quali un sonar o un laser scanner e

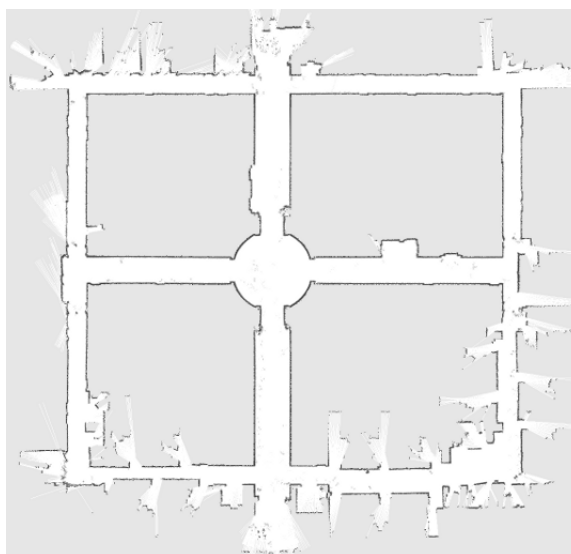


Figura 1.4: Esempio di mappa a griglia: le celle bianche rappresentano spazio libero, quelle nere spazio occupato; quelle grigie sono indefinite.

possono contribuire alla fase di pianificazione del percorso del robot tramite una procedura di associazione che calcola la trasformazione necessaria per allineare due mappe, riducendo gli errori di pianificazione e localizzazione. I principali vantaggi di questo approccio sono la possibilità di ottenere mappe precise e con risoluzioni abbastanza elevate anche di ambienti grandi, e la facilità di memorizzazione e aggiornamento.

In alcuni casi [2] le mappe a griglia sono costruite utilizzando approcci derivati dall'intelligenza artificiale, dove la mappa dell'ambiente può essere nota a priori e viene utilizzata per l'addestramento di una rete neurale. Durante il movimento del robot la rete genera le distribuzioni di probabilità delle celle, che vengono integrate per aggiornare la mappa. Nel caso di mappe di grandi dimensioni alcuni lavori, [3], utilizzano metodi probabilistici per stimare la migliore mappa ottenibile e permettere contemporaneamente la localizzazione del robot. Il problema della costruzione della mappa in questo caso viene formulato in termini di massimizzazione di una funzione di *likelihood*, rispettando i vincoli geometrici che descrivono l'ambiente e il movimento del robot. La fase di localizzazione viene facilitata dalla presenza di landmark nell'am-

biente, che in questo caso devono essere identificati manualmente, mentre in altri casi essi vengono identificati tramite procedure automatiche che si occupano anche della localizzazione del robot.

### 1.2.2 Mappe di poligonali

Nella mappe composte da poligonali gli oggetti presenti nell'ambiente sono modellati tramite archi poligonali, o semplicemente poligonali, cioè insiemi ordinati di punti connessi da segmenti, che sono adatte per essere impiegate con sensori di prossimità ad elevata risoluzione come il laser scanner. Questo tipo di mappe consente di descrivere con maggior dettaglio l'ambiente osservato, rispetto ad altri tipi di features, e nel contempo permette di rappresentare gli oggetti in modo semplice e generale.

Le principali operazioni da eseguire nella costruzione della mappa riguardano l'estrazione e segmentazione delle scansioni ed in particolare l'associazione di poligonali proveniente da una scansione e dalla mappa corrente per ottenere una nuova poligonale come risultante della fusione delle poligonali di partenza. Alcuni approcci [4], [5] non fanno uso di informazioni odometriche durante la costruzione della mappa rendendo più robusta la rappresentazione anche in caso di slittamenti del robot o errori di misurazione dei sensori.

Le poligonali estratte dalle scansioni rappresentano il blocco fondamentale di questo tipo di mappe e permettono di facilitare la fase di associazione rispetto ad altri tipi di rappresentazioni perchè contengono una quantità maggiore di informazione e possono essere raggruppate, unite o fuse tra di loro. In [7] vengono esposti alcuni metodi per la fusione di poligonali, con lo scopo di ridurre il numero di features nella mappa e quindi eliminare le informazioni ridondanti. Questo risultato si ottiene tramite la fusione di più poligonali che rappresentano lo stesso oggetto in una sola. I metodi descritti in [7] si distinguono in: fusione di punti collineari su un segmento o linea, fusione di coppie di segmenti e fusione di insiemi di segmenti, cioè poligonali che formano il contorno di un oggetto.



Figura 1.5: Esempio di mappa a poligonali (immagine tratta da [4]).

In [4] viene mostrato un altro metodo per la fusione che identifica le singole poligonali nella scansione e ricerca gli oggetti corrispondenti nella mappa. Questi oggetti vengono poi fusi tra di loro in un singolo oggetto la cui posizione o livello di dettaglio sono stati migliorati dalla fusione avvenuta. L'algoritmo segue l'ordine dei punti che compongono la poligonale e simula il funzionamento del laser scanner: nel caso in cui un raggio simulato intersechi la poligonale viene creato un nuovo punto chiamato *simulated scan point*. In più vengono aggiunti anche i punti finali della poligonale all'insieme di nuovi punti simulati creato. Successivamente per ogni punto di questo insieme viene trovato il punto più vicino sulla mappa e, se la distanza è minore di una certa soglia, viene calcolata una media pesata tra i due punti. Il punto medio così ottenuto viene inserito nella nuova mappa. Se non vengono trovati vicini entro la soglia prestabilita il punto simulato viene inserito come nuovo nella mappa e rappresenta una nuova features poligonale che non era stata osservata prima.

Un altro metodo di fusione tra poligonali viene descritto in [5], dove viene

mostrato come costruire una mappa globale fondendo scansioni consecutive, senza l'uso di informazioni di posizione, ma solo tramite le relazioni geometriche tra gli elementi nelle scansioni. Il metodo presentato si articola nell'integrazione di scansioni consecutive in una mappa parziale e, successivamente, nella fusione di più mappe parziali in una mappa globale. Questo sistema è vantaggioso dal punto di vista della possibilità di raccogliere i dati sull'ambiente in momenti o sessioni diverse e con più robot che cooperano, che non necessitano di sapere la loro posizione reciproca.

### 1.2.3 Mappe di features

Nella mappe di features l'ambiente viene rappresentato da alcune caratteristiche particolari, o *features*, cioè elementi dell'ambiente che possono essere stabilmente riconosciuti nei dati sensoriali, anche in diversi istanti di tempo durante l'esplorazione del robot, e che possono essere rappresentati da vettori di parametri. Solitamente la feature è associata ad una porzione di spazio che presenta un alto contenuto informativo: alcuni esempi di features possono essere spigoli, angoli, muri o altro. Le caratteristiche vengono estratte da diversi tipi di sensori, come sonar, laser scanner o telecamere.

Negli ultimi anni è sempre più diffuso l'uso di tecniche di visione artificiale nella robotica, ed è applicato a diversi tipi di ambienti e di robot. La ricchezza di informazione fornita da un sensore di visione permette di estrarre un gran numero di features e di utilizzarle per la costruzione di mappe, per la localizzazione e per la navigazione. Un caso specifico di applicazione di una tecnica di localizzazione basata su feature riguarda [13], dove sono state utilizzate le features SIFT (*Scale-Invariant Feature Transform*) [14] per localizzare globalmente il robot. Queste features sono state sviluppate per il riconoscimento di oggetti in applicazioni di visione artificiale, e sono invarianti alle operazioni di traslazione, scala, e rotazione effettuate sull'immagine, oltre ad essere poco sensibili ai cambi di illuminazione e di inquadratura. Quindi ad ogni feature SIFT sono associate la posizione nell'immagine, la scala e l'orientazione. Questi parametri consentono di usare le SIFT come

features per procedure di SLAM robuste e computazionalmente leggere. La posizione globale del robot viene ottenuta tramite l'associazione di features estratte dal frame corrente con quelle contenute in una mappa, che viene utilizzata come un database. L'algoritmo crea delle sottomappe in tre dimensioni accumulando una serie di frames consecutivi, e applica una procedura di allineamento con la mappa generale per migliorare e irrobustire la fase di localizzazione. Inoltre le sottomappe vengono fuse insieme per generare un mappa 3D dell'ambiente, considerando i casi di chiusura del loop che permettono di correggere l'errore di posizione accumulato in tutti gli allineamenti delle precedenti sottomappe.

In un altro caso, descritto in [15], l'idea principale del lavoro svolto consiste nel stimare la probabilità che due osservazioni siano state acquisite nella stessa zona dell'ambiente, per risolvere alcuni problemi nella navigazione e localizzazione come la chiusura del loop. Per far questo è stato sviluppato un framework Bayesiano per la localizzazione e il mapping che fa uso di feature basate sulle SIFT, che rappresentano il mondo come una lista di attributi sull'apparenza della scena, scelti da un ampio insieme chiamato vocabolario. La mappa è costituita da un insieme di zone distinte, parametrizzate dalla distribuzione di probabilità generata dalle features osservate. Questo modello delle osservazioni consente di localizzare il robot e può essere aggiornato in tempo lineare con il numero delle zone presenti nella mappa. Di conseguenza le prestazioni di questo algoritmo permettono di individuare i casi di chiusura del loop in linea.

Esistono anche altri tipi di features, come per esempio FLIRT [16], cioè features multiscala adatte per operare con sensori di prossimità, e quindi anche laser scanner o sonar.

Uno dei primi lavori che fa uso di features per il sonar è [17], in cui viene creata la mappa dell'ambiente e viene usato un filtro di Kalman esteso per la procedura di localizzazione. Le osservazioni estratte dal sonar vengono confrontate con delle features contenute nella mappa come muri, angoli o superfici cilindriche. La fase iniziale del procedimento consiste nell'acquisi-

zione di alcuni landmark le cui posizioni sono note prima del movimento del robot. In un secondo tempo la strategia adottata in questo approccio cerca di eliminare l'incertezza associata ad un insieme di osservazioni, e successivamente aggiorna la mappa con le osservazioni corrette. Questo contribuisce a semplificare la stima dello stato del sistema attraverso il filtro di Kalman e semplifica la procedura di associazione delle features.

### 1.3 Costruzione incrementale di mappe locali

La costruzione incrementale di una mappa necessita di metodi e algoritmi per l'integrazione dei dati provenienti dai sensori con i dati che sono stati immagazzinati nella mappa stessa nei passi precedenti. In questo processo la fase di associazione tra le features della scansione e quelle della mappa è fondamentale per mantenere la coerenza della rappresentazione. Considerando l'incertezza sulla posizione attuale del robot e la possibilità di avere un ambiente dinamico, dove gli oggetti presenti nella scena possono avere posizioni diverse anche tra scansioni successive, l'associazione diventa ancora più critica. Un problema nell'associazione infatti può portare al mancato riconoscimento degli oggetti presenti e anche alla costruzione di una mappa imprecisa o errata. Viceversa una corretta esecuzione di questa fase permette di identificare gli oggetti osservati precedentemente, e questo consiste in un importante fattore discriminante per classificare un oggetto come statico o dinamico.

La conoscenza della posa del robot al momento dell'osservazione permette di creare mappe locali più precise, dato che fornisce una stima utile a posizionare correttamente gli elementi della mappa. Nel caso in cui sia costruita una mappa su larga scala è necessario impiegare dei metodi di risoluzione della mappa stessa, alcuni dei quali sono stati introdotti in 1.1, per mantenere consistente la rappresentazione.

Per poter effettuare l'associazione di elementi geometrici è necessario disporre di una metrica per il confronto fra gli elementi stessi, che fornisca una stima

della loro distanza reciproca o similarità, e di una procedura per selezionare le coppie di elementi che saranno confrontate utilizzando la metrica scelta. In letteratura sono presenti diverse metriche introdotte in tempi relativamente recenti, alcune delle quali provengono dal contesto dell'*image processing*. La metrica descritta in [28] è efficiente, deterministica, orientata alla percezione parziale e relativamente robusta a rumore e trasformazioni degli oggetti in esame. Il calcolo di questa metrica è preceduto da un filtraggio sui dati grezzi, chiamato *Discrete Curve Evolution* (DCE) [29], che riduce l'influenza del rumore e semplifica la forma della feature senza eliminare i dettagli rilevanti. Gli oggetti vengono poi descritti tramite il loro contorno, interpretato come un poligono chiuso, ricavato direttamente dalla segmentazione delle immagini di partenza. I contorni vengono a loro volta scomposti in archi convessi o concavi che vengono raggruppati secondo il loro ordine. Successivamente viene definita una funzione tangente che dato un insieme di archi calcola un istogramma degli angoli associati, che viene usato per ottenere una misura della similarità, tramite confronto delle parti più significative degli oggetti. Il procedimento è riassunto in figura 1.6. Un approccio simile è descritto in [30],

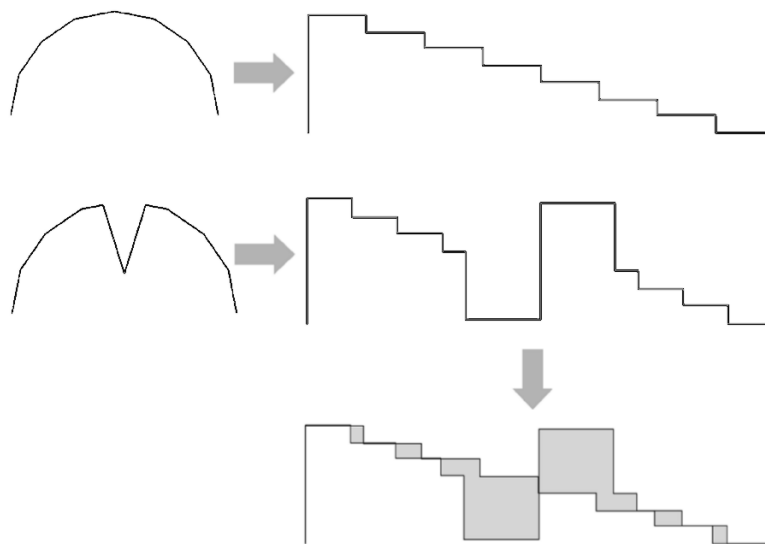


Figura 1.6: Similarità tra due poligoni: calcolo delle funzioni tangente e della loro distanza (immagine tratta da [28]).

---

dove la funzione tangente viene utilizzata su tutta la forma della poligonale. Questa metrica è applicabile anche a casi in cui i contorni degli oggetti sono solo parzialmente visibili o quando hanno forme complicate.

## Capitolo 2

# Requisiti del problema e scelte progettuali

Questo capitolo presenta i principali componenti del sistema realizzato, che riguardano in particolare gli elementi scelti per la rappresentazione dell'ambiente e gli algoritmi utilizzati per la creazione e l'aggiornamento della mappa. Per comprendere le caratteristiche e il funzionamento in dettaglio di questi aspetti è necessario descrivere preventivamente il problema affrontato in questo lavoro di tesi.

### 2.1 Descrizione del problema

Gli LGV impiegati in ambito industriale sono normalmente equipaggiati con sensori laser scanner per evitare collisioni e, più in generale, garantire la sicurezza nei movimenti. In particolare, il rilevamento di persone o oggetti è comunemente effettuato definendo alcune aree del piano di scansione e verificando la presenza di ostacoli al loro interno. La figura 2.1 mostra un esempio schematico della configurazione utilizzata più frequentemente. Questo metodo risulta in molti casi concreti poco flessibile se l'ambiente circostante subisce cambiamenti a medio termine, per esempio quando oggetti quasi-statici (fardelli, pallet, attrezzatura, ecc.) vengono collocati in regioni

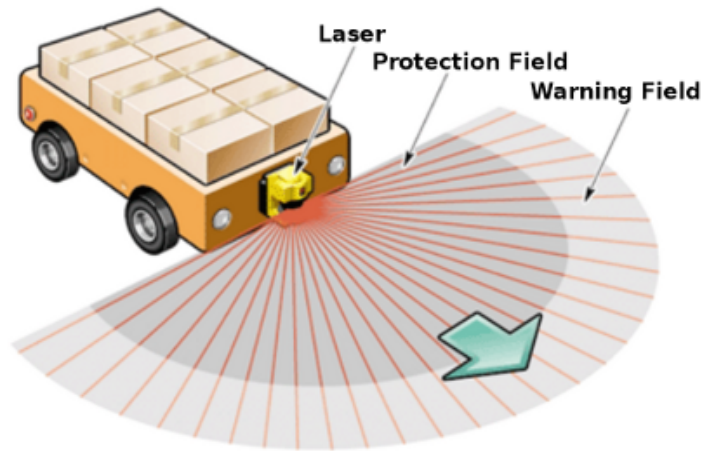


Figura 2.1: Aree di sicurezza di un veicolo LGV dotato di un sensore laser.

che inizialmente erano libere. Gli oggetti quasi-statici vengono definiti come oggetti che generano percezioni sensoriali da parte del robot anche per intervalli di tempo prolungati e che non possono essere considerati statici, perchè non fanno parte dell'infrastruttura dell'ambiente di navigazione, ma nemmeno dinamici perchè la loro posizione è fissa per intervalli di tempo brevi, paragonabili al tempo necessario per il transito del robot e per l'acquisizione delle scansioni.

Le aree definite sul sensore determinano il comportamento del veicolo e quindi sono critiche per il buon funzionamento degli LGV, che è caratterizzato principalmente, negli impianti di stoccaggio, dal tempo con cui vengono eseguite le missioni di navigazione e posizionamento delle merci. Infatti in presenza di rilevazioni sensoriali all'interno delle aree di sicurezza il veicolo deve rallentare oppure fermarsi a seconda della situazione rilevata, ovvero della zona in cui sono stato rilevati ostacoli di qualche tipo. Ciò determina una riduzione dell'efficienza operativa rispetto a quanto previsto in assenza di ostacoli.

## 2.2 Condizioni operative critiche

Nelle installazioni sul campo la rilevazione di percezioni non previste nelle aree di sicurezza da parte dell'LGV può essere dovuta a diverse situazioni:

- la presenza di un operatore;
- la presenza di una navetta, automatizzata o a guida manuale, in transito in un'area in cui essa non è prevista;
- modifiche “permanenti” apportate all'ambiente dopo l'installazione del sistema di LGV (spostamento di scaffalature o simili);
- la presenza di oggetti quasi-statici di qualsiasi natura lasciati in prossimità delle zone di transito degli LGV.

Possono essere definite diverse politiche di gestione di queste situazioni, in base alla loro gravità e alla distanza dall'ostacolo individuato. Nel primo caso una politica corretta impone al veicolo di rallentare o di fermarsi per garantire il livello di sicurezza previsto dalle normative. Nel terzo caso una modifica permanente apportata all'ambiente dovrebbe determinare anche una modifica delle traiettorie degli LGV o un ridisegno delle aree di sicurezza definite sui sensori. Il quarto caso è quello di maggior interesse per l'attività svolta in questa tesi e descrive il problema di cui ci si è occupati. Questo problema può essere per esempio dovuto a una scatola o un altro oggetto che viene lasciato per ore a margine dell'area di transito degli LGV e per tutto questo periodo ne determina un rallentamento. Un operatore umano si renderebbe conto della natura del problema, e presumibilmente fin dal primo passaggio procederebbe a piena velocità, non sussistendo un reale problema di sicurezza. Un LGV, viceversa, allo stato attuale rileva la presenza dell'oggetto ai bordi dell'area di transito, definita anche come “warning field” in figura 2.1, e continua a rallentare durante tutti i passaggi in quella zona fino a che l'oggetto permane.

## 2.3 Riconoscimento di oggetti quasi-statici e dinamici

Scopo di questo progetto è sviluppare un metodo affidabile, efficiente ed economico per identificare oggetti quasi-statici, in modo da consentire a chi sviluppa il sistema di controllo e navigazione degli LGV di adottare le politiche di gestione di tale evento che ritiene più appropriate. É possibile delineare almeno due tipologie di approccio al problema della identificazione mediante scansioni laser di oggetti quasi-statici o dinamici. Entrambi verranno descritti nel seguito di questa sezione.

### 2.3.1 Accumulo di scansioni

Un primo approccio prevede l'analisi di scansioni consecutive o comunque acquisite a breve distanza di tempo dal laser scanner, identificando mediante una elaborazione locale i dati percettivi che cambiano tra le due scansioni. In tal modo è possibile classificare le entità percepite in posizioni diverse tra le due scansioni come “dinamiche”, e presumibilmente associate a persone o a veicoli in movimento. Quindi in questo primo caso l'individuazione di oggetti classificabili come quasi-statici è affidata alla rilevazione di dati sensoriali variabili associati alle entità presenti nella zona di scansione del laser scanner. La figura 2.2 mostra un esempio di rilevamento di un oggetto in movimento tramite il confronto di tre scansioni consecutive. Questo approccio può essere ritenuto non sufficientemente sicuro in particolari condizioni, come la presenza di un operatore che si mantiene immobile per qualche istante; inoltre gli oggetti quasi-statici potrebbero essere classificati come statici in un mappa ottenuta dall'accumulo di poche scansioni, dando luogo a falsi negativi che potrebbero compromettere le prestazioni del sistema e soprattutto la sicurezza della navigazione.

In letteratura questo approccio è definito in alcuni lavori: per esempio in [31] viene sviluppato un modello in grado di rappresentare le parti statiche e dinamiche dell'ambiente. La soluzione proposta utilizza un modulo unico

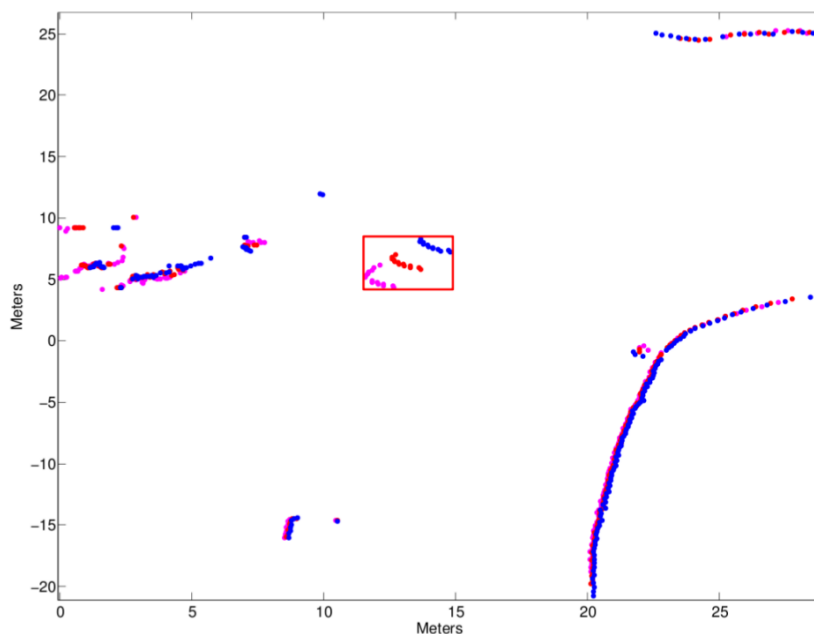


Figura 2.2: Confronto di scansioni consecutive per determinare le parti in movimento (immagine tratta da [32]).

che si occupa sia del problema di SLAM, attraverso una mappa a griglia che descrive parti statiche usata successivamente per le fasi di pianificazione della traiettoria e di rilevazione degli ostacoli, sia del riconoscimento di oggetti dinamici, tramite una serie di filtri di Kalman, uno per ogni oggetto, che tracciano la loro posizione. I filtri sono indipendenti tra di loro, e il loro vettore di stato è composto dalla posizione e dalla velocità stimata dell'oggetto dinamico associato. Viene ipotizzata una velocità costante per gli oggetti e per le persone, e nonostante queste ultime possano fare movimenti particolarmente rapidi e imprevedibili, il modello è sufficiente per il tracciamento di tutti i tipi di movimenti nella maggior parte delle situazioni. In più il sistema deve creare e rimuovere i filtri a seconda degli oggetti che entrano o escono dalla scena, e suddividere i filtri in modo che ognuno sia associato ad un solo oggetto. Il modulo sviluppato viene poi integrato nella struttura robotica di navigazione in cui l'uso selettivo delle informazioni sugli oggetti dinamici nella fase di pianificazione permette di evitare comportamenti non

desiderati del robot in situazioni complicate, come ostruzioni del passaggio o oggetti in movimento diretti verso il robot.

Questo metodo raggruppa gli oggetti in due grandi gruppi (o cluster), statici o dinamici, mentre il metodo proposto in [32] consiste in un raggruppamento parzialmente supervisionato di osservazioni sensoriali riguardanti oggetti dinamici secondo il loro tipo di movimento. Viene quindi presentato un framework probabilistico per la classificazione e il tracciamento di oggetti mobili. Il framework è basato sui *Conditional Random Fields* (CRF) [33], un modello probabilistico a grafo originariamente creato per la classificazione di dati sequenziali. Il metodo proposto è in grado di stimare il moto degli oggetti e di lavorare anche con stime iniziali affette da incertezza e in casi in cui siano presenti sovrapposizioni tra i punti delle scansioni che rappresentano gli oggetti dinamici. Inoltre vengono considerate le relazioni spaziali tra punti vicini nella scansione per diminuire gli errori commessi nella classificazione, mentre le relazioni tra oggetti in movimento contribuiscono a definire i cluster in cui gli oggetti vengono suddivisi. Questo sistema consente di determinare i cluster e i corrispondenti pattern di moto, in modo da raggruppare gli oggetti dinamici e, se necessario, inizializzare un filtro per il tracciamento della loro posizione.

### 2.3.2 Confronto di sottomappe

Una seconda tipologia di approccio per il riconoscimento di oggetti quasi-statici prevede di confrontare le scansioni laser acquisite in un certo tratto percorso dal robot con quelle acquisite in occasione del precedente transito nel medesimo tratto. In questo secondo caso si ipotizza che il transito del robot avvenga a intervalli minimi dell'ordine di alcuni secondi o più, e pertanto la probabilità che un oggetto o una persona mantengano la medesima identica posa tra i due passaggi può essere ragionevolmente considerata trascurabile. Infatti la configurazione degli oggetti quasi-statici non cambia rapidamente e essi vengono osservati come statici per brevi intervalli di tempo. Si ipotizza quindi che i diversi stati o configurazioni di questi oggetti saranno osservati

solo nei transiti successivi del robot nella stessa zona. Lo svantaggio di questo approccio è che esso non consente di classificare entità percepite come oggetti quasi-statici già nel primo transitto nel tratto in questione. Per le applicazioni di cui questo progetto si occupa si ritiene che questo svantaggio sia trascurabile rispetto alla maggiore sicurezza offerta dalla metodologia adottata.

In [34] viene descritto un metodo per modellare esplicitamente le parti dinamiche di un ambiente, in modo da mantenere nella mappa anche i dettagli relativi agli aspetti dinamici, che spesso sono rilevanti per la fase di localizzazione. L'idea principale di questo approccio consiste nel raccogliere informazioni sui cambiamenti osservati nell'ambiente e stimare le possibili configurazioni spaziali degli oggetti, le quali verranno rappresentate nella mappa. In particolare il metodo si concentra sugli oggetti quasi-statici che appaiono

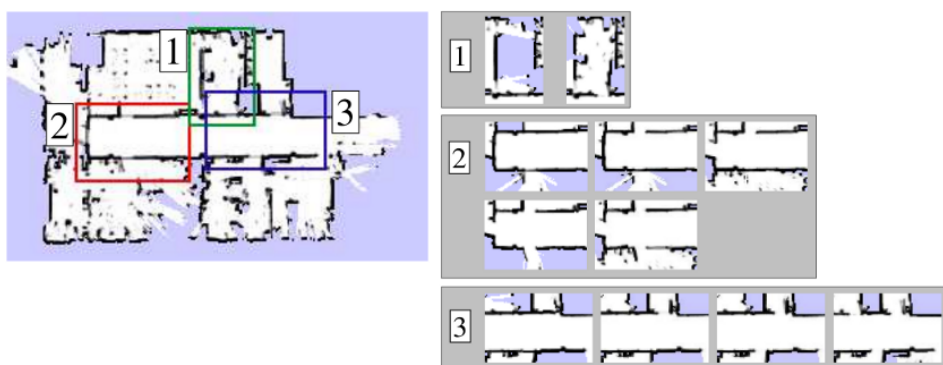


Figura 2.3: Diverse configurazioni degli oggetti apprese dal sistema per la sottomappa di sinistra (immagine tratta da [34]).

solo in un numero limitato di configurazioni e posizioni, e apprende queste possibili configurazioni durante il passaggio del robot. Nel sistema vengono prima identificate le zone in cui le osservazioni sensoriali indicano la presenza di oggetti in movimento, e in seguito l'area interessata viene allargata considerando i vincoli di visibilità tra le aree, fino a raggiungere una superficie massima fissata. Questo limite sulla dimensione massima riduce il numero di oggetti dinamici per ogni sottomappa e di conseguenza la complessità del problema. Per mezzo di questi dati raccolti viene costruita una sottomappa

locale, a griglia, per ogni zona dell'ambiente dove sono stati osservati oggetti quasi-statici. Le sottomappe così ottenute vengono poi classificate per ottenere una base di conoscenza con tutti i possibili stati dell'ambiente. La figura 2.3 mostra una sottomappa e le relative configurazioni estrapolate dal sistema per ogni zona di interesse.

In [35] viene presentato un approccio intermedio tra quelli descritti in questa sezione e nella precedente. Esso consiste in un framework di localizzazione in grado di individuare e tracciare la posa del robot anche in ambienti dinamici. Il metodo si basa sulle osservazioni sensoriali causate da oggetti quasi-statici e su una mappa degli oggetti statici nota a priori per stimare la posa del robot. Gli oggetti dinamici, cioè quelli che variano la loro posizione durante l'acquisizione dei dati, vengono identificati e rimossi, in modo da considerare solo i cambiamenti che avvengono nell'ambiente in modo lento. Vengono costruite delle sottomappe locali, che hanno una validità limitata nel tempo, per ogni nuovo oggetto quasi-statico individuato che non fa parte della mappa statica conosciuta. Quando il robot acquisisce nuove informazioni su una zona dell'ambiente, la corrispondente sottomappa può essere aggiornata se ritenuta non più affidabile. La nuova sottomappa consente di integrare le informazioni contenute nella mappa statica con le osservazioni aggiornate sugli oggetti quasi-statici individuati. In questo modo si costruisce una mappa estesa, un esempio è mostrato in figura 2.4, che rappresenta un modello esplicito sia degli oggetti statici sia di quelli quasi-statici e viene confrontato con le osservazioni correnti del robot per determinare una mappa di riferimento utile nella stima della posa. Se le osservazioni corrispondono a oggetti presenti nella mappa statica questa viene usata come riferimento, altrimenti viene utilizzata la sottomappa locale più vicina al robot, che rappresenta gli oggetti quasi-statici osservati. La fase di localizzazione, basata su filtri particellari, utilizza la mappa estesa per cercare di risolvere i problemi dovuti all'occlusione degli oggetti statici e all'impossibilità di osservare caratteristiche dell'ambiente troppo lontane. Infatti questo metodo permette di ottenere prestazioni migliori rispetto ad altri metodi di localizzazione,

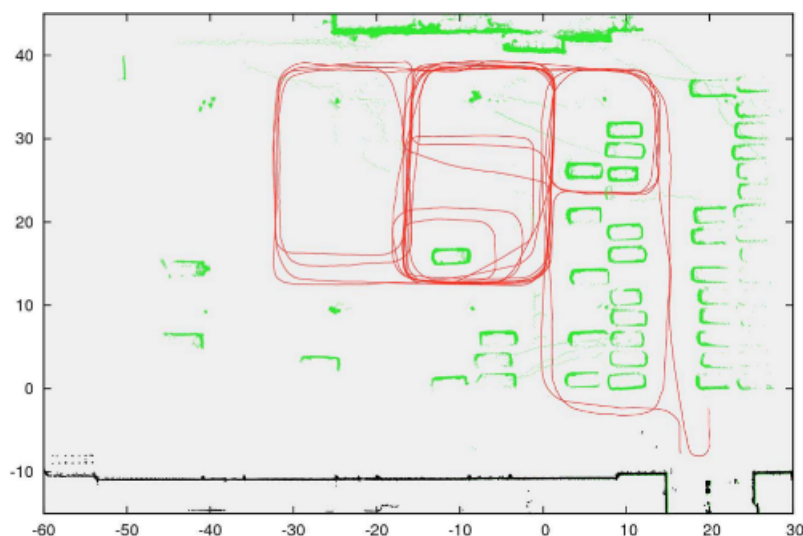


Figura 2.4: Mappa estesa composta da oggetti statici (neri) e oggetti quasi-statici (verdi) che sono usati per migliorare le prestazioni di localizzazione del sistema in presenza di variazioni nell'ambiente (immagine tratta da [35]).

per esempio quelli basati su *Rao-Blackwellized Particle Filter* (RBPF) [36] o *Graph-SLAM* [37], soprattutto quando applicato in spazi grandi, dove il robot può percorrere lunghi tratti e gli oggetti della mappa statica possono essere lontani e quindi non osservabili da una certa posizione. Di conseguenza, in queste situazioni, gli oggetti quasi-statici osservabili rappresentano informazioni importanti per la localizzazione, che non vengono considerate negli approcci citati, mentre in questo caso permettono di ottenere errori di localizzazione bassi anche in ambienti dinamici.

## 2.4 Soluzione per l'individuazione di oggetti quasi-statici

L'approccio utilizzato per individuare gli oggetti quasi-statici prevede di confrontare le scansioni laser acquisite in un certo tratto percorso dall'LGV con quelle acquisite in occasione del precedente transito nel medesimo tratto. Questo metodo garantisce una buona affidabilità nel riconoscimento degli

oggetti e inoltre può essere utilizzato senza dotare gli LGV di ulteriori sensori, ma integrando le informazioni provenienti dai laser scanner già montati sui veicoli. Questa strategia privilegia inoltre la sicurezza della identificazione rispetto ad altri criteri, per cui è estremamente improbabile che una persona possa essere rilevata erroneamente come oggetto quasi-statico. La figura 2.5

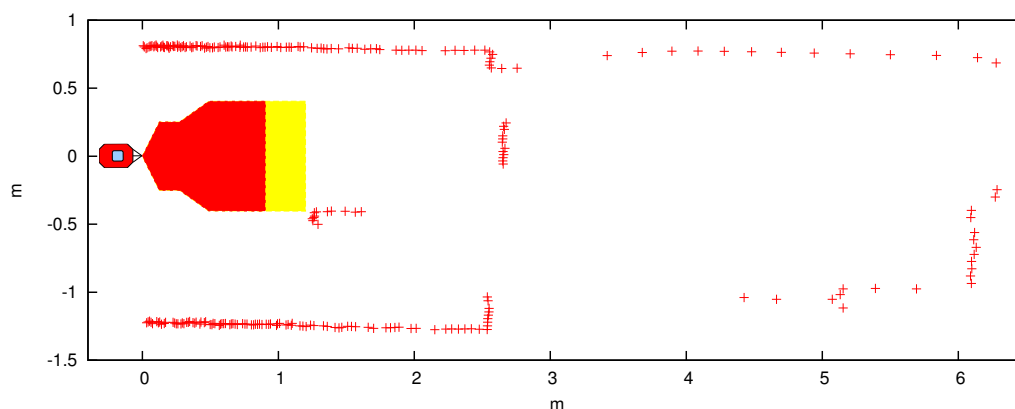


Figura 2.5: Scansione ottenuta in un corridoio con alcuni ostacoli sulla traiettoria del robot. L'area di sicurezza più esterna è segnata in giallo, mentre l'area di protezione più interna è disegnata in rosso.

mostra una scansione laser come esempio di discussione. Se l'LGV opera in condizioni normali tutti i punti della scansione saranno collocati al di fuori delle aree di sicurezza. La presenza di punti della scansione all'interno delle aree di sicurezza richiede una classificazione del dato sensoriale e la adozione di specifiche politiche di navigazione: per quanto riguarda questo lavoro di tesi si ipotizza che l'LGV sia in grado di muoversi autonomamente su traiettorie predefinite, calcolate in base al layout disponibile dell'impianto. Non è richiesta la presenza di una mappa dell'impianto nota a priori per il corretto funzionamento del sistema. Nella soluzione proposta si ipotizza inoltre che le traiettorie siano suddivise in brevi segmenti (dell'ordine di pochi metri al massimo), che l'LGV percorre in successione per completare la missione corrente. Tutte le immagini sensoriali acquisite dal veicolo all'interno di un segmento di traiettoria sono riferite alla prima posa del robot in quel segmento. Il sistema cioè costruisce un insieme di rappresentazioni parziali di

ciò che viene percepito tramite il laser scanner nelle aree di sicurezza, mentre non costruisce alcuna rappresentazione globale. Tra le zone corrispondenti a segmenti consecutivi del percorso dell'LGV possono essere presenti, in generale, delle parziali sovrapposizioni. Tali sovrapposizioni rendono il sistema più robusto anche se parzialmente ridondante. La scelta di non rappresentare i segmenti in riferimento ad un frame globale semplifica l'elaborazione e la costruzione delle mappe, in particolare per quanto riguarda l'associazione tra dati sensoriali acquisiti nella scansione corrente e dati già presenti nella mappa.

La classificazione di una rilevazione sensoriale come oggetto statico o quasi-statico deve essere eseguita in linea, durante il moto dell'LGV, e deve avvenire in tempi tali da consentire al robot di compiere qualunque azione necessaria al mantenimento dei requisiti di sicurezza. Infatti, una volta classificato come quasi-statico un oggetto presente in un'area di sicurezza ma non di transito, l'LGV può proseguire a piena velocità oppure ad una velocità intermedia, ma in tutti i casi questa è una decisione che può essere adottata dallo sviluppatore e può essere una politica modificabile a seconda dell'impianto o della situazione in cui ci si trova. L'identificazione di oggetti quasi-statici potrebbe anche portare, se ritenuto utile, alla segnalazione di tale evento in un pannello operatore per favorirne la rimozione in caso di presenza impropria. In più, in caso di modifiche permanenti dell'ambiente che involontariamente vanno ad interferire con le aree di sicurezza, la rilevazione della configurazione di oggetti quasi-statici può aiutare a correggere il layout dell'impianto.

## 2.5 Rappresentazione degli oggetti

Uno dei principali requisiti della soluzione da proporre consiste nel ridurre al minimo la codifica impiegata dalla mappa per consentire l'eventuale trasmissione della rappresentazione degli ostacoli ad altri veicoli dell'impianto. La rappresentazione adottata deve offrire soluzioni algoritmiche compatibili con le capacità di elaborazione disponibili nel contesto applicativo, e deve garan-

tire l'affidabilità e la conformità ai requisiti di sicurezza, in particolare delle persone. Inoltre, la raccolta di scansioni in modo cumulativo impone l'uso di una rappresentazione che sia facilmente integrabile con la mappa esistente e che renda possibile il riconoscimento di oggetti quasi-statici. La rappresentazione scelta deve inoltre descrivere l'ambiente con sufficiente dettaglio senza essere ridondante, e deve permettere di utilizzare un metodo di associazione tra scansioni consecutive veloce e preciso.

È stato scelto di rappresentare l'ambiente tramite linee spezzate, a cui si farà riferimento in seguito con il nome di poligonali o polilinee, composte da un insieme ordinato di segmenti consecutivi: dati  $n + 1$  punti  $v_0, \dots, v_n$  si definisce poligonale la curva costituita dall'unione dei segmenti  $s_i = v_{i-1} v_i$  con  $i = 1, \dots, n$ . Si ipotizza che la poligonale sia una curva semplice, cioè che l'intersezione di due segmenti sia vuota se i segmenti non sono consecutivi  $s_i \cap s_j = \emptyset$  per  $j \neq i + 1$ , oppure contenga un solo punto se i segmenti sono consecutivi nell'ordine scelto  $s_i \cap s_j = \{v_i\}$  per  $j = i + 1$ . Questo punto viene chiamato vertice. La figura 2.6 mostra un esempio di poligonale. Le poli-

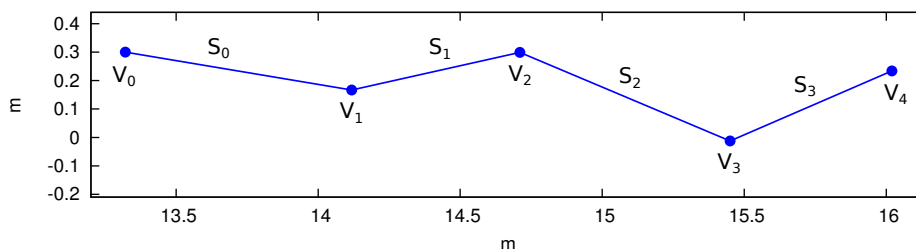


Figura 2.6: Esempio di linea spezzata costituita da segmenti ordinati e vertici, chiamata poligonale.

gonali rappresentano l'elemento base della mappa, e possono essere estratte dalla scansione corrente, combinate insieme per formare nuove poligonali, semplificate tramite algoritmi che mantengono la forma della poligonale riducendone il numero di vertici, e integrate nella mappa locale presente nel sistema, come descritto nel capitolo 3.

La fase di estrazione fa uso di algoritmi specifici per l'analisi di scansioni che restituiscono la lista di poligonali estratte, ordinata secondo il senso di scan-

sione del sensore laser durante l'acquisizione. Nella maggior parte dei casi le poligonali estratte da una singola scansione sono sufficienti ad identificare gli oggetti che ricadono nelle aree di sicurezza. Durante la fase di integrazione ogni poligonale della scansione corrente viene associata con la poligonale più simile nella mappa, ed eventualmente ne modifica la conformazione per rispecchiare i cambiamenti nell'ambiente in cui il robot si muove. Nelle map-

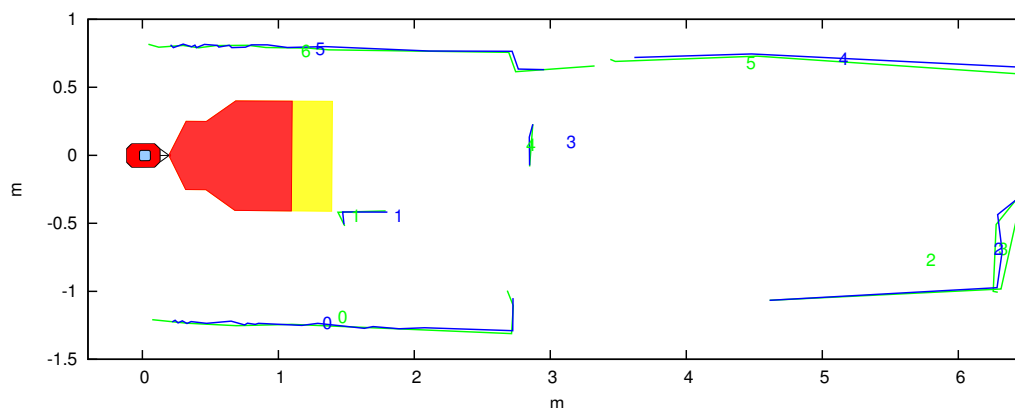


Figura 2.7: Poligonali estratte dalla scansione laser corrente (blu) e ottenute dalla fusione delle poligonali presenti nella mappa e derivanti dalle precedenti scansioni (verde). A scopo illustrativo le poligonali in figura sono estratte dall'intera scansione e non limitandosi ai punti all'interno delle aree di sicurezza.

pe create vengono considerate solo le poligonali che intersecano almeno una delle aree programmate sul laser, per simulare la situazione di lavoro degli LGV e dei sensori laser con cui sono equipaggiati. In più questo permette di limitare il numero di poligonali di cui è costituita la mappa e quindi limitare il tempo di esecuzione richiesto dall'algoritmo, mantenendo il livello di dettaglio richiesto per l'applicazione. Quindi, siccome ciascuna poligonale è rappresentata tramite la lista ordinata dei suoi vertici, la rappresentazione può essere archiviata con limitata occupazione di memoria e trasmessa ad altri LGV anche nel caso in cui il canale di comunicazione utilizzato abbia una ampiezza di banda ridotta.

In figura 2.7 sono raffigurate le poligonali estratte dalla scansione di figura 2.5 dove, per motivi illustrativi, sono presenti tutte le poligonali e non solo

quelle che risiedono in una area.

La mappa locale così ottenuta risulta però parziale a causa di occlusioni e meno robusta in presenza di dati incerti e rumorosi. Tali difetti possono essere attenuati fondendo le poligonali estratte da scansioni acquisite consecutivamente. Inoltre, la fusione di poligonali risulta necessaria quando siano presenti più sensori laser scanner sullo stesso LGV. La figura 2.7 mostra inoltre il risultato della fusione di più scansioni consecutive, il cui procedimento verrà illustrato nel capitolo successivo.

## Capitolo 3

# Algoritmi per la costruzione di mappe locali di poligoni

La necessità di fondere le informazioni ricavate da scansioni consecutive, acquisite in posizioni diverse dall'LGV in movimento, impone di studiare le modalità di elaborazione delle scansioni. L'elaborazione si può suddividere in diverse fasi, che vengono elencate e poi verranno descritte in dettaglio nel seguito:

- *estrazione*: vengono estratte le poligoni dalla scansione corrente;
- *associazione*: le poligoni estratte vengono associate a quelle già presenti nella mappa;
- *fusione*: le poligoni della scansione vengono fuse con quelle della mappa, sulla base delle associazioni trovate al passo precedente, ed il risultato è un nuovo insieme di poligoni;
- *unione*: le poligoni nella mappa vengono ordinate secondo i loro angoli di visibilità, e successivamente i punti terminali che si trovano ad una distanza minore di una soglia vengono uniti per creare una nuova poligonale.

### 3.1 Estrazione di poligoni

La prima elaborazione svolta sui dati grezzi ricavati dai sensori consiste in un filtraggio mirato a rimuovere gli outliers della scansione corrente. Il filtro si basa sulla distanza geometrica dei punti e sulle proprietà statistiche di questa distanza. Viene calcolato un vettore di distanze, dove l'elemento  $i$ -esimo rappresenta la distanza tra il punto  $i$  e il punto  $i - 1$  nella scansione. Successivamente vengono calcolate media e deviazione standard degli elementi del vettore contenuti in una finestra mobile, di dimensione fissa, centrata nell'elemento  $i$ -esimo. Il punto  $i$ -esimo della scansione viene filtrato se la sua distanza dal precedente non rientra nell'intervallo di confidenza dato dalla deviazione standard calcolata, cioè se è verificata la seguente condizione:

$$|distance_i - mean_i| > C \cdot stddev_i \quad (3.1)$$

dove *distance* è il vettore delle distanze, *mean* e *stddev* sono i vettori delle statistiche sulla finestra mobile,  $C$  è un coefficiente per impostare la confidenza voluta in percentuale sulla deviazione standard. La figura 3.1 mostra un esempio di outlier che viene rimosso dal criterio descritto in precedenza. Una volta completato questo primo filtraggio sui dati i punti rimanenti vengono analizzati da un algoritmo di suddivisione, che identifica le discontinuità tra punti successivi all'interno della scansione per suddividere i punti in gruppi, tramite due soglie adattative per le distanze tra i punti. A questo passo dell'algoritmo la scansione è stata scomposta in una lista di intervalli di punti contigui. La figura 3.2 mostra una scansione e i segmenti che potrebbe essere estratti dopo l'ultima elaborazione effettuata. Ciascun intervallo è un insieme di punti che rappresenta un curva quasi continua. L'obiettivo di questa fase è l'estrazione di una poligonale da questa curva, i cui vertici corrispondono alle variazioni di orientamento principale. Per questo è necessario elaborare ulteriormente gli intervalli calcolati nella prima fase per ottenere una ricostruzione più precisa e fedele dell'ambiente. In particolare ogni intervallo viene elaborato mediante l'algoritmo di Ramer-Douglas-Peucker [42], che elimina i punti non necessari per rappresentare con sufficiente li-

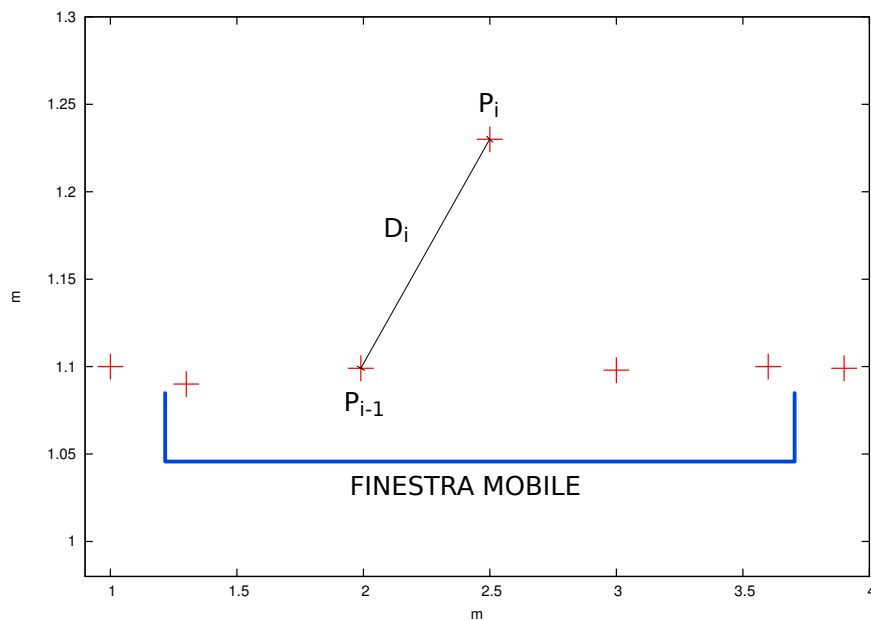


Figura 3.1: Il punto  $P_i$  viene rimosso dalla scansione in quanto la sua distanza  $D_i$  non ricade nell'intervallo di confidenza imposto dalle statistiche sui punti inclusi nella finestra mobile.

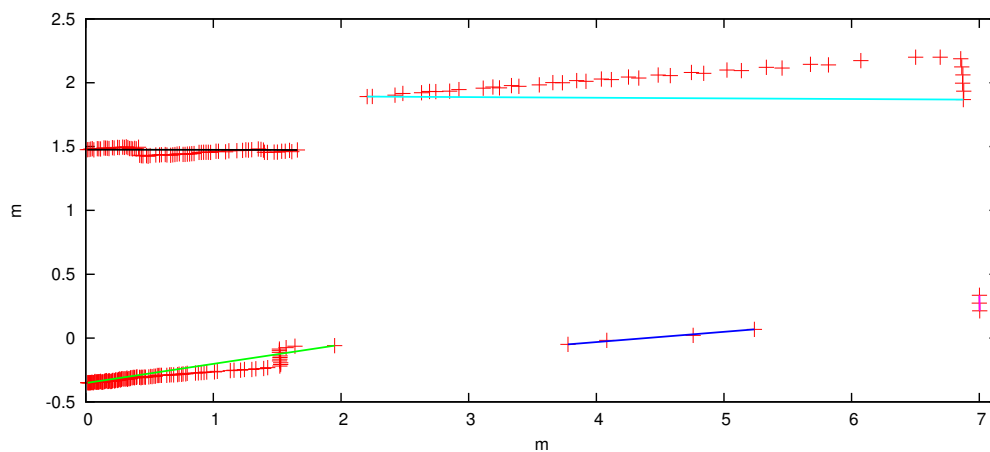


Figura 3.2: Segmenti estratti da una scansione dopo la suddivisione iniziale in insiemi di punti disgiunti.

vello informativo l'ambiente scansionato. Infatti questo algoritmo suddivide l'intervallo in ulteriori sottointervalli secondo il criterio della collinearità, per definire in modo preciso i contorni degli oggetti.

Le poligoni risultanti da questo procedimento rappresentano i bordi degli oggetti con un numero molto inferiore di punti, quindi possono essere facilmente condivise tra diversi sistemi e confrontate tramite i loro vertici. La figura 3.3 mostra le poligoni ottenute dalla stessa scansione di figura 3.2: si nota come le poligoni ricoprono in modo fedele le parti della scansione, rispecchiando la struttura dell'ambiente acquisita tramite il sensore. La

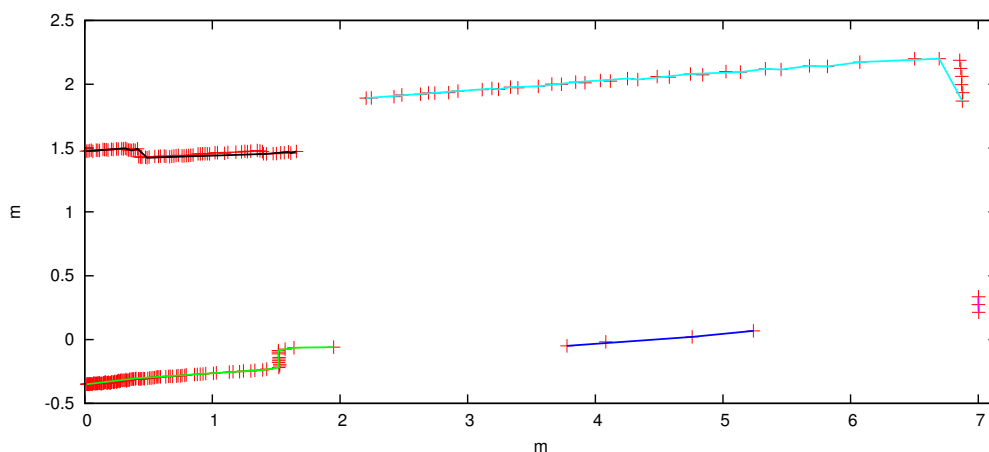


Figura 3.3: Poligoni estratte partendo dagli insiemi di punti disgiunti trovati nella prima fase, e elaborate per aumentarne la precisione nella ricostruzione dell'ambiente.

procedura di estrazione è necessaria per evitare che parti non connesse della scansione (per esempio contorni di oggetti diversi) risultino appartenenti alla stessa poligonale, generando quindi dei contorni fittizi che non rispecchiano la struttura dell'ambiente. La suddivisione così ottenuta consente anche di facilitare la fase di associazione che si basa su caratteristiche geometriche delle poligoni, che non sarebbero utilizzabili nel caso di poligoni contenenti l'intero insieme di punti della scansione.

## 3.2 Associazione

L'associazione di poligoni appartenenti ad insiemi diversi è l'operazione più importante dell'algoritmo di costruzione di una mappa, e quindi costituisce una fase critica per la buona riuscita delle fasi successive. Infatti, una corretta esecuzione dell'associazione permette di costruire mappe consistenti e, soprattutto, di identificare gli oggetti osservati o non osservati precedentemente. Si possono individuare i seguenti aspetti del problema:

- l'associazione avviene tra gli elementi di un insieme, la scansione corrente, e gli elementi di un altro insieme, la mappa locale ottenuta dall'algoritmo con le scansioni acquisite fino al passo precedente;
- è necessario disporre di una metrica per il confronto fra gli elementi da associare;
- occorre una procedura per selezionare le coppie di elementi che saranno confrontate utilizzando la metrica scelta.

La posizione relativa delle poligoni dei due insiemi può essere calcolata con la sola odometria, oppure impiegando una stima più accurata della posizione del robot, come descritto nel capitolo 4.

La fase di associazione ha lo scopo di determinare relazioni di vicinanza spaziale e di somiglianza tra le poligoni estratte nella scansione corrente e quelle già presenti nella mappa locale. È necessario trovare la migliore associazione possibile tra i due insiemi di dati, preservando l'ordinamento e permettendo anche associazioni multiple.

### 3.2.1 Similarità

Prima di applicare un algoritmo di risoluzione di questo problema è necessario definire la metrica utilizzata per il confronto tra poligoni, cioè stimare la similarità tra gli elementi da associare. Questa metrica è stata proposta in tempi relativamente recenti e si basa sulle metriche descritte in 1.3: è

efficiente, deterministica, orientata alla percezione parziale e relativamente robusta a variazioni del punto di osservazione. La metrica considera alcune caratteristiche geometriche estratte dalle poligoni stesse, e deve permettere di riconoscere poligoni simili nonostante queste non siano identiche o siano ruotate tra di loro. In particolare si basa sulla forma descritta dai segmenti della poligonale: per ciascuna delle poligoni viene costruito un istogramma angolare che codifica le informazioni sulla forma: a ciascun segmento della poligonale, opportunamente normalizzato rispetto alla lunghezza della poligonale stessa, viene associato l'angolo compreso fra l'asse delle ascisse e il segmento.

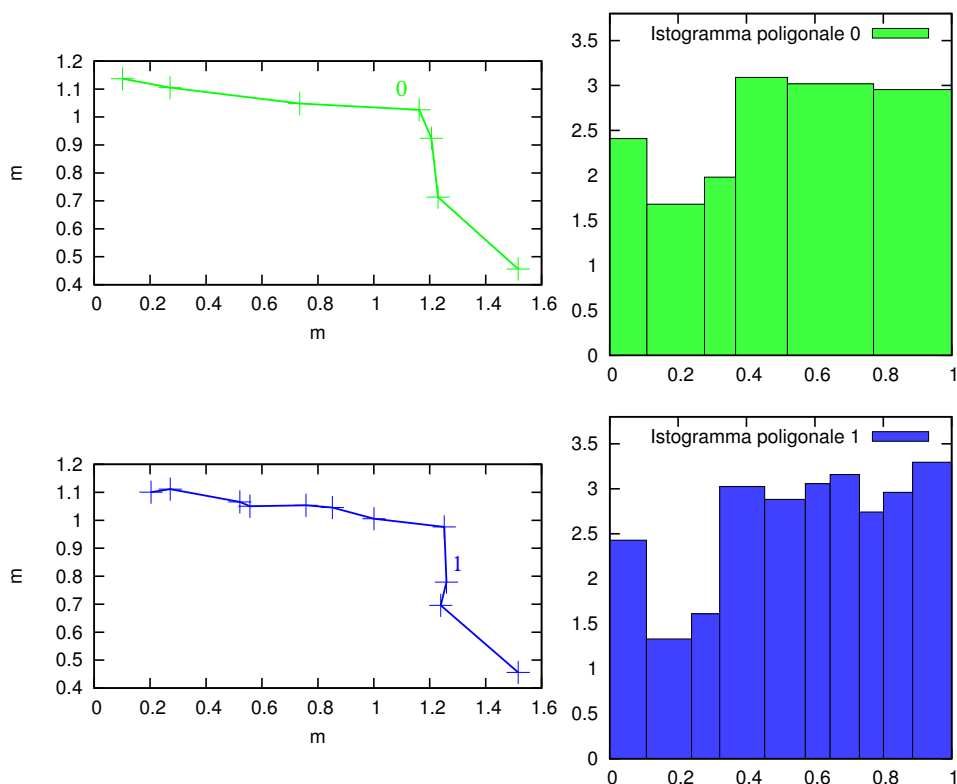


Figura 3.4: Istogramma di una poligonale: ogni elemento dell'istogramma rappresenta l'angolo compreso fra un segmento della poligonale e l'asse delle ascisse. Le figure rappresentano due poligoni (sinistra) ed i corrispondenti istogrammi (destra) impiegati nel calcolo della similarità.

La figura 3.4 mostra una coppia di poligoni con i corrispondenti istogrammi. La distanza di similarità tra due poligoni viene calcolata come la differenza tra gli istogrammi associati alle poligoni. L'istogramma associato ad una poligonale può essere rappresentato con una funzione tangente; ogni poligonale  $P$  viene parametrizzata da una ascissa curvilinea normalizzata  $s \in [0, 1]$ , dove  $P(s)|_{s=0}$  indica il primo vertice della poligonale, mentre  $P(s)|_{s=1}$  indica l'ultimo vertice. La funzione tangente  $T_P(s) : [0, 1] \mapsto [0, 2\pi]$  associa ad ogni punto della poligonale  $P$  il corrispondente valore angolare,  $\varphi$ , come mostrato anche in figura 3.5, a sinistra.

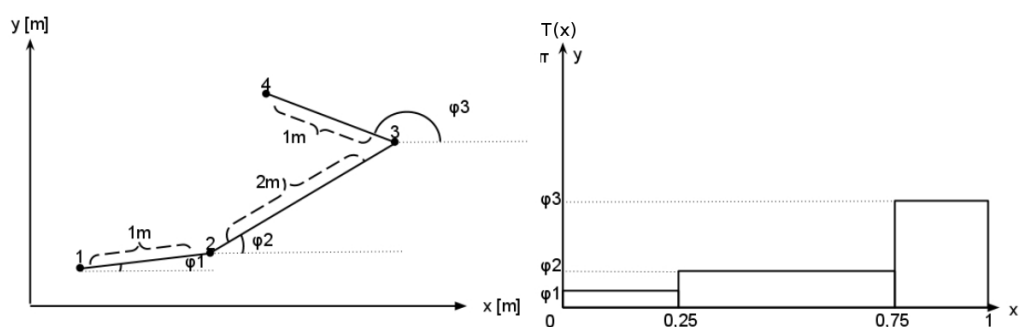


Figura 3.5: Funzione tangente di una poligonale: esempio di calcolo e grafico

La normalizzazione rispetto alla lunghezza della poligonale permette il confronto di un istogramma con altri istogrammi applicando una funzione di similarità. Esistono varie definizioni per questa funzione, una delle più utilizzate [6] è mostrata in 3.2 per due poligoni  $P$  e  $D$

$$S(P, D) = \left( \int_0^1 (T_P(s) - T_D(s) + \theta_0)^2 ds \right) \cdot \max \left[ \frac{l(P)}{l(D)}, \frac{l(D)}{l(P)} \right] \quad (3.2)$$

$$\theta_0 = \int_0^1 T_P(s) - T_D(s) ds \quad (3.3)$$

dove  $T_P$  e  $T_D$  sono le funzioni tangente delle poligoni  $P$  e  $D$ ,  $l(\cdot)$  denota la lunghezza di una poligonale, e  $\theta_0$  è l'angolo medio di rotazione che massimizza la similarità tra gli istogrammi associati alle singole poligoni. Il termine che moltiplica l'integrale introduce un peso significativo nel caso la lunghezza delle poligoni sia notevolmente diversa: in questo modo viene considerata la

dimensione della poligonale oltre alla forma, che può risultare discriminante in casi particolari, dato il tipo di sensore e l'algoritmo di estrazione utilizzati. Questa metrica risulta conveniente da usare dal punto di vista algoritmico in quanto il suo calcolo risulta notevolmente semplificato dal tipo di funzione (a gradino). Un esempio del risultato ottenuto con due poligonali è mostrato in figura 3.6.

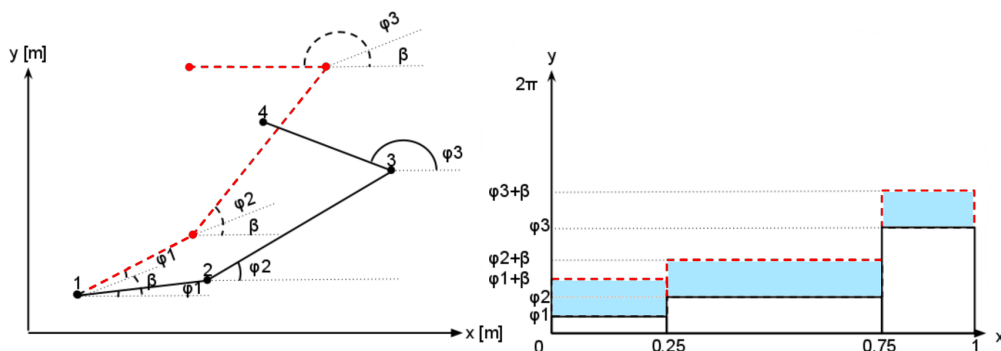


Figura 3.6: Funzioni tangente di due poligonali (a sinistra) e distanza tra gli istogrammi associati alle poligonali, rappresentata dall'area azzurra (a destra).

L'area colorata rappresenta la differenza tra i due istogrammi analizzati e, graficamente, il risultato dell'integrazione nella formula 3.2 per il calcolo della somiglianza. La successiva fase di associazione si basa su questa misura per determinare se due poligonali appartenenti a due insiemi diversi debbano essere associate.

### 3.2.2 Procedura di associazione

Nell'effettuare il confronto fra due insiemi di poligonali, l'algoritmo adottato considera l'ordinamento delle poligonali rispetto al punto di osservazione, quello cioè dal quale è stata acquisita la scansione corrente. L'algoritmo inizia sempre dalla prima poligonale visibile dal primo beam di destra e procede in senso antiorario. L'ordine di osservazione previene inconsistenze nella fase di associazione.

Il problema consiste nel confrontare due liste di poligoni e trovare le migliori corrispondenze tra gli elementi, minimizzando la distanza complessiva calcolata su tutti gli elementi associati tramite la metrica descritta precedentemente. La figura 3.7 mostra le associazioni ottenute tra due insiemi di poligoni, che per scopi illustrativi sono state estratte da due scansioni vicine ma non consecutive. Si tratta quindi di un problema di associazione multi-a-molti che viene risolto mediante tecniche di *programmazione dinamica*, (dynamic programming), le quali permettono di ottenere la soluzione finale suddividendo il problema in sottoproblemi più semplici. L'idea fonda-

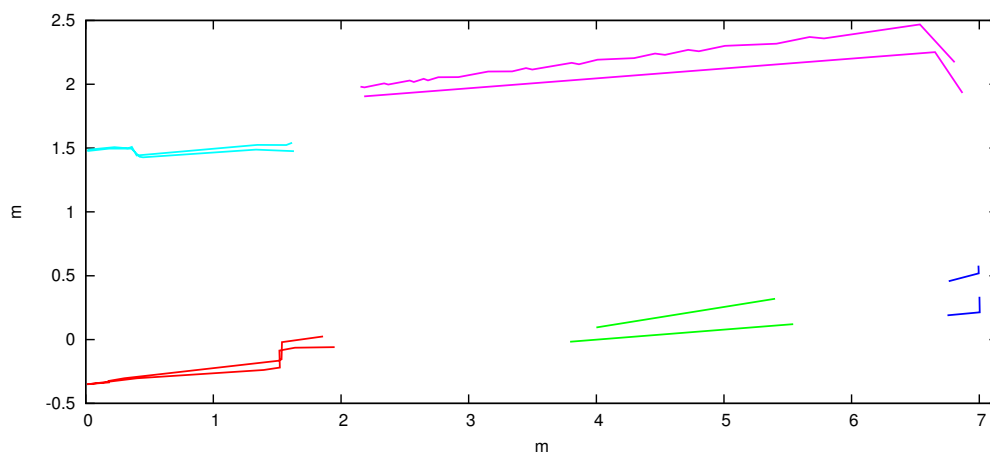


Figura 3.7: Poligoni estratte da due scansioni vicine e associate: i colori rappresentano gli abbinamenti scelti dall'algoritmo.

mentale è quella di costruire una matrice in cui ogni cella, con indici  $(i, j)$ , rappresenta la soluzione migliore del sottoproblema definito dall'associazione delle prime  $i$  poligoni del primo insieme e delle prime  $j$  poligoni del secondo insieme. La soluzione ottima del problema  $(i, j)$  viene calcolata valutando le soluzioni ottime dei problemi  $(i - 1, j - 1)$ ,  $(i - 1, j)$  e  $(i, j - 1)$ . Quindi nella generica cella  $(i, j)$  viene inserita la soluzione del sottoproblema corrispondente, considerando valida l'associazione dell'arco  $i$ -esimo e dell'arco  $j$ -esimo:

$$M_{i,j} = S(i, j) + \min(M_{i-1,j}, M_{i,j-1}, M_{i-1,j-1}) \quad (3.4)$$

dove  $S$  denota la funzione metrica di distanza tra due poligoni descritta precedentemente. Il termine minimo calcolato in 3.4 rappresenta il valore minimo delle celle adiacenti, e garantisce che solo la soluzione migliore (cioè quella minima) di ogni sottoproblema venga propagata alle celle successive. Un caso particolare è rappresentato dalla prima riga e dalla prima colonna della matrice, che devono essere preventivamente inizializzate: ogni cella  $(i, j)$  contiene il minimo tra la cella precedente (la cella  $(i, j - 1)$  sulla riga  $i$ , o la cella  $(i - 1, j)$  sulla colonna  $j$ ) e la distanza tra gli elementi  $i$  e  $j$ , cioè  $S(i, j)$ . Dopo aver valutato tutte le coppie di elementi che compongono la matrice è possibile ottenere la soluzione ottima del problema partendo dalla cella in basso a destra e ripercorrendo la matrice verso l'alto scegliendo il percorso che include il valore minimo tra le celle adiacenti. Gli indici delle celle toccate durante questa procedura rappresentano gli accoppiamenti ottimi tra gli elementi da associare.

La procedura permette anche di associare un arco di un insieme a più archi dell'altro insieme, costruendo quindi una o più associazioni multiple. Questo permette di risolvere alcuni problemi derivanti da errori nell'estrazione delle poligoni: infatti può accadere che un oggetto venga rappresentato da una sola poligonale in una vista e da due poligoni in una visuale diversa, derivata da un'altra scansione. Questo può succedere nei casi in cui ci siano occlusioni nella scansione, oppure quando l'orientazione del sensore è tale da avere misurazioni distanziate fra di loro, che possono essere suddivise in più poligoni durante la fase di estrazione. Queste situazioni possono dar luogo a matrici di associazione non quadrate che vengono correttamente interpretate dall'algoritmo.

### 3.3 Filtraggio

Uno svantaggio relativo a questa misura di similarità consiste nella scarsa rilevanza data alla posizione relativa delle poligoni, nel valutarne le associazioni. Pertanto le corrispondenze ottenute secondo il criterio di similarità

durante la fase di associazione vengono successivamente valutate calcolando la loro distanza di Hausdorff modificata, che deriva dalla distanza tra segmenti ed è adattata alle poligoni. La distanza funge da elemento validatore e quindi realizza un filtraggio ulteriore, che elimina eventuali associazioni non corrette. Siano  $P_i$  e  $P_j$  le due poligoni associate, la loro distanza viene calcolata come:

$$D(P_i, P_j) = \min \left\{ \max_{r \in P_j} d(P_i, r), \max_{s \in P_i} d(P_j, s) \right\} \quad (3.5)$$

dove la funzione  $d(P_i, r)$  calcola la distanza minima del punto  $r \in P_j$  da tutti i segmenti formati da coppie di punti consecutivi in  $P_i$ . In figura 3.8, è mostrato schematicamente il calcolo delle distanze  $d(P_1, r_j)$ ,  $j = 1, 2, 3$  e  $d(P_2, s_i)$ ,  $i = 1, 2, 3$  per i vertici delle poligoni  $P_1$  e  $P_2$ . Successivamente per completare il calcolo della distanza utilizzata,  $D(P_1, P_2)$ , è necessario scegliere il massimo valore tra le distanze  $d(P_1, r_j)$  e tra le distanze  $d(P_2, s_i)$  per le due poligoni, e infine il minimo tra i due valori ottenuti. Il filtraggio

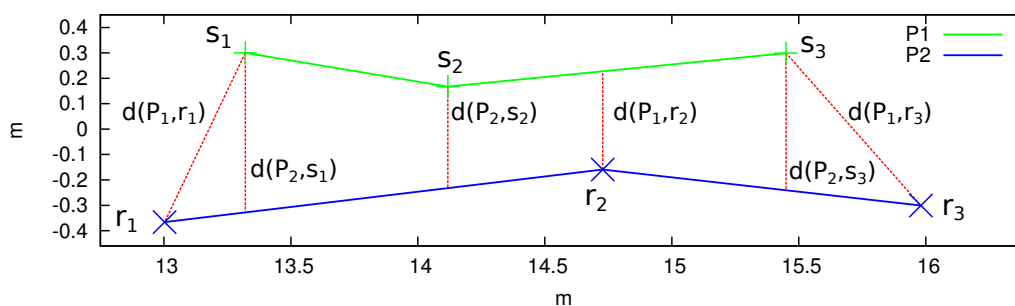


Figura 3.8: Calcolo della distanza  $D$ , di equazione 3.5, tra due poligoni.

opera eliminando associazioni tra due poligoni quando la loro distanza è più grande di una soglia prestabilita. Questo filtraggio è utile per rimuovere associazioni tra poligoni che si somigliano come forma, ma che sono troppo distanti fra di loro. La dipendenza di questa procedura da una soglia la rende molto sensibile ai cambiamenti di infrastruttura dell'ambiente, ed in particolare il valore della soglia deve essere adattato quando variano le dimensioni dell'ambiente, per garantire associazioni corrette.

### 3.4 Fusione di poligoni

Questa fase dell'algoritmo di creazione della mappa consiste nel fondere le poligoni estratte dalla scansione corrente con le corrispondenti poligoni che si trovano già nella mappa. In questo modo il sistema mantiene aggiornata la mappa durante l'acquisizione dei dati. La procedura di fusione fa uso delle associazioni calcolate al passo precedente, e effettua una proiezione della poligonale appartenente alla scansione sulla poligonale associata nella mappa. Quindi il processo restituisce una sola poligonale partendo da una coppia, operando vertice a vertice e cercando di semplificare la struttura della poligonale risultante senza perdere i dettagli che ne determinano la forma. Denominate  $P_1$  la poligonale della mappa,  $P_2$  la poligonale della scansione corrente e  $P_i^j$  il  $j$ -esimo punto della poligonale  $i$ , l'algoritmo di fusione proietta i punti di  $P_2$  su  $P_1$  trattando opportunamente i casi particolari derivanti da un numero diverso di punti. Durante questa operazione viene preservato

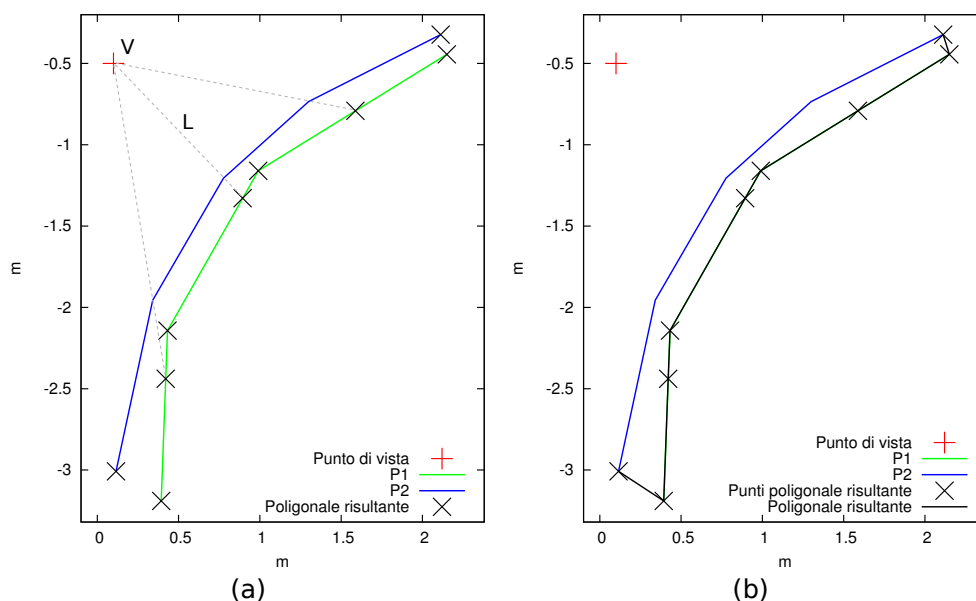


Figura 3.9: Esempio di fusione di due poligoni: in nero i punti della poligonale risultante dalla fusione di  $P_1$  e  $P_2$  (a) e la poligonale completa (b).

l'ordinamento dei punti in senso antiorario rispetto ad punto di osservazione.

Nel caso della costruzione di una mappa di poligoni tale punto coincide con il sistema di riferimento posizionato al centro del sensore laser.

Un esempio di fusione è mostrato in figura 3.9: per semplicità sono raffigurate solo due poligoni, ma la fusione può avvenire anche su insiemi contenenti più poligoni, che sono state associate secondo la procedura descritta in 3.2. Nel caso in cui l'ordinamento tra i punti  $V$ ,  $P_1^i$  e  $P_2^j$  di figura 3.9 non sia antiorario l'algoritmo calcola tutte le intersezioni tra la linea passante per  $V$  e per il punto  $P_2^j$ , e tutti i segmenti della poligonale  $P_1$ . L'intersezione più vicina a  $V$  viene aggiunta come punto della poligonale in uscita, e permette di mantenere l'ordinamento dei vertici della poligonale in senso antiorario rispetto a al punto di osservazione.

Per cercare di controllare gli effetti della fusione sulle poligoni della mappa è stata sviluppata una versione modificata dell'algoritmo, che consente di inserire nella mappa il punto medio tra  $P_2^i$  e l'intersezione più vicina a  $V$  calcolata secondo la tecnica descritta in precedenza. Attraverso il calcolo della

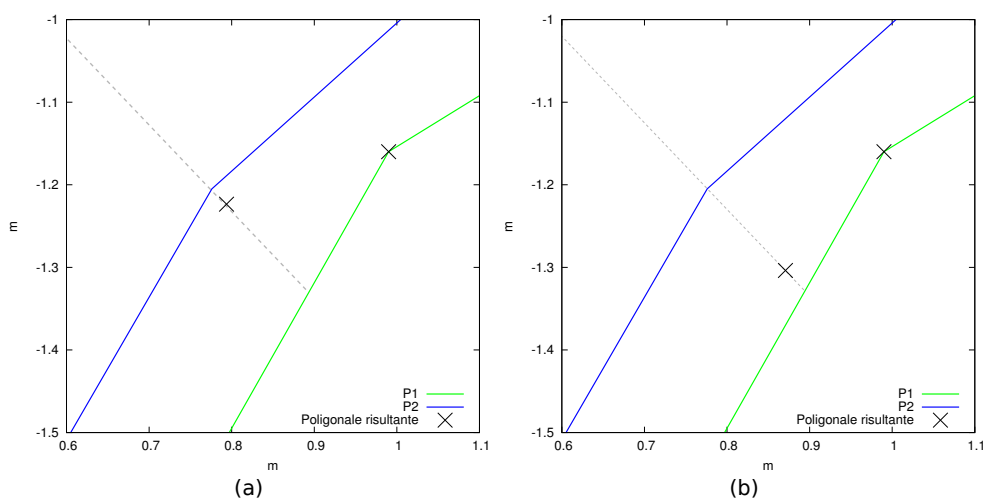


Figura 3.10: Risultato della fusione di una scansione con la mappa. (a) Il punto inserito nella mappa è vicino alla scansione quindi si ha una modifica significativa alla poligonale corrispondente, (b) la modifica della mappa è ridotta dal valore dei pesi, siccome il punto inserito si trova vicino alla poligonale della mappa stessa.

media pesata, si può accentuare la modifica della mappa tramite la fusione

con la scansione corrente, oppure è possibile ridurre gli effetti della fusione ed effettuare solo modifiche marginali alla mappa, scegliendo opportunamente i valori dei pesi. La figura 3.10 mostra un confronto dei nuovi punti ottenuti come intersezioni dalla procedura di fusione, con pesi molto diversi.

L'algoritmo 3.1 descrive con maggiore dettaglio le fasi della procedura di fusione tra due poligoni.

**Algoritmo 3.1** Fusione tra due poligoni.

---

**Input:**  $P_1$  poligonale di riferimento;

**Input:**  $P_2$  poligonale da proiettare;

**Input:**  $V$  punto di vista;

**Output:**  $P_{out}$  poligonale risultante;

```

1:  $i \leftarrow 0, j \leftarrow 0$ ;
2: while  $i < P_1.size$  or  $j < P_2.size$  do
3:   if  $i \geq P_1.size$  then
4:      $P_{out} \leftarrow P_{out} + P_2^j$ ;
5:      $j \leftarrow j + 1$ ;
6:   else if  $j \geq P_2.size$  then
7:      $P_{out} \leftarrow P_{out} + P_1^i$ ;
8:      $i \leftarrow i + 1$ ;
9:   else
10:    if  $V, P_1^i, P_2^j$  sono in disposti in senso antiorario then
11:       $P_{out} \leftarrow P_{out} + P_1^i$ ;
12:       $i \leftarrow i + 1$ ;
13:    else
14:       $L \leftarrow$  linea passante per  $V$  e  $P_2^j$ ;
15:      for  $c = 1 \rightarrow P_1.size$  do
16:         $S \leftarrow$  segmento di poligonale  $[P_1^{c-1}, P_1^c]$ ;
17:         $I_c \leftarrow$  intersezione tra  $L$  e  $S$ ;
18:      end for
19:      if almeno una intersezione then
20:         $I_{min} \leftarrow$  intersezione tra i punti  $I_c$  più vicina a  $V$ ;
21:         $P_{out} \leftarrow P_{out} + I_{min}$ 
22:      else
23:         $P_{out} \leftarrow P_{out} + P_2^j$ 
24:      end if
25:       $j \leftarrow j + 1$ ;
26:    end if
27:  end if
28: end while

```

---

### 3.5 Unione di poligoni

La modalità secondo cui avviene l'aggiornamento della mappa, durante il movimento del robot, introduce alcuni problemi nella fusione di poligoni in casi particolari. Per esempio quando un oggetto di grandi dimensioni viene inquadrato a lungo dal sensore mentre il robot transita nelle vicinanze può accadere che vengano create più poligoni che corrispondono però allo stesso oggetto. La procedura di fusione illustrata nel paragrafo precedente permette di fondere archi poligonali sovrapposti, ma non esclude la presenza di poligoni parzialmente sovrapposti, o con estremità molto vicine. La figura 3.11 mostra proprio uno di questi casi, in cui le poligoni  $P_1$  e  $P_2$  rappresentano due tratti del contorno dello stesso oggetto, che di conseguenza sarebbero convenientemente unite in una sola poligonale per mantenere la consistenza dei dati nella mappa. Un beneficio secondario consiste nella limitazione del numero di poligoni presenti, dal quale dipende la complessità delle diverse fasi dell'algoritmo di costruzione.

La procedura di unione viene eseguita periodicamente, ogni  $n$  scansioni acquisite, e ha il compito di collegare fra di loro le poligoni presenti nella mappa che hanno estremità adiacenti, e che sono adiacenti anche nell'ordine di visibilità all'interno della mappa stessa. Questo ordinamento facilita la gestione della mappa, in particolare nell'associazione e fusione di poligoni. La distanza viene misurata, in questo caso, da una metrica che calcola la distanza minima tra le estremità delle poligoni, e che consente di individuare oggetti vicini senza considerare altre proprietà come la forma che nel caso dell'unione non sono rilevanti. L'algoritmo di unione, infatti, è generale e può essere utilizzato con qualunque coppia di poligoni: la risultante potrà essere successivamente semplificata se necessario, secondo la tecnica descritta in [42].

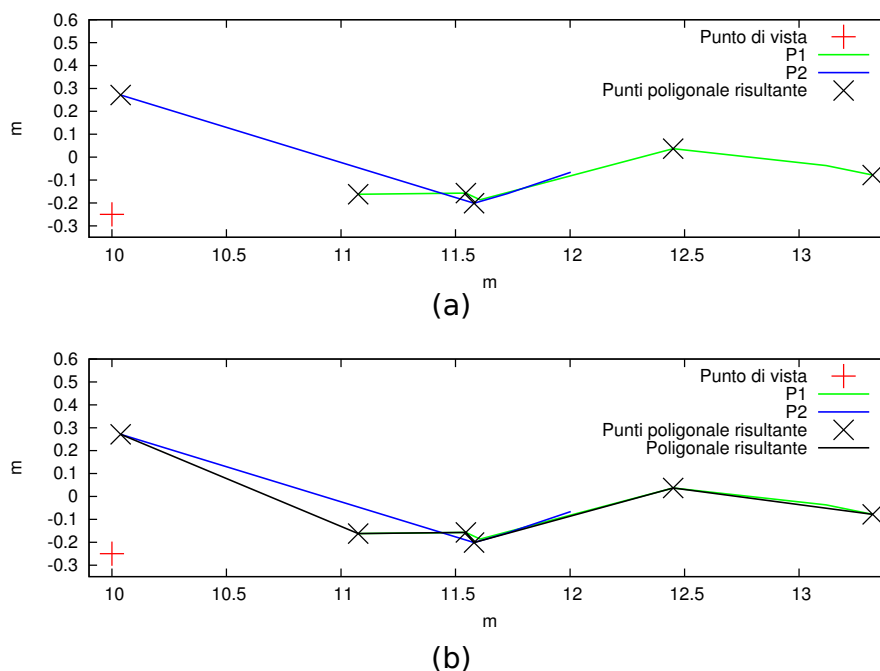


Figura 3.11: Esempio di unione di due poligoni: in nero i punti della poligonale risultante dall'unione di  $P_1$  e  $P_2$  (a) e la poligonale completa (b).

### 3.5.1 Riordino delle poligoni

Oltre all'ordinamento dei vertici interni, le poligoni sono ordinate nell'ambito della scansione laser da cui sono ricavate, secondo la loro visibilità, riferita rispetto ad un punto di vista stabile che rappresenta il sistema di riferimento dal quale è stata acquisita la scansione corrente. L'ordine individuato nella scansione viene mantenuto anche nella mappa, durante tutte le fasi di associazione, fusione e aggiornamento, tramite una procedura di ordinamento che viene utilizzata periodicamente, e in particolare viene richiamata dopo ogni esecuzione dell'algoritmo di unione. L'ordinamento è mantenuto anche nel caso in cui una poligonale estratta dalla scansione rappresenti un nuovo oggetto, e quindi non venga associata con quelle presenti nella mappa. La nuova poligonale viene inserita nella mappa nella posizione corretta, dopo aver calcolato la sua visibilità.

La figura 3.12 mostra una costruzione geometrica utile per definire la visi-

bilità di una poligonale riferita all'origine degli assi. Gli angoli  $\alpha$  e  $\beta$  sono calcolati considerando tutti i vertici della poligonale, e costituiscono il massimo e minimo angolo di visuale che si ottiene definendo un punto di vista  $V$  nel piano della scansione, secondo le seguenti equazioni:

$$\alpha = \max_{p \in P} \left\{ \arctan \frac{p_y - V_y}{p_x - V_x} - V_\theta \right\} \quad (3.6)$$

$$\beta = \min_{p \in P} \left\{ \arctan \frac{p_y - V_y}{p_x - V_x} - V_\theta \right\} \quad (3.7)$$

dove  $P$  rappresenta una poligonale,  $p$  un vertice della poligonale e  $V_\theta$  è l'orientamento del punto di vista espresso rispetto a un sistema di riferimento globale.

Gli angoli di visibilità sono sempre compresi nell'intervallo  $[-\pi; \pi]$ , ed è sufficiente scegliere indifferentemente uno di questi angoli per ordinare tutte le poligonali attorno a  $V$ .

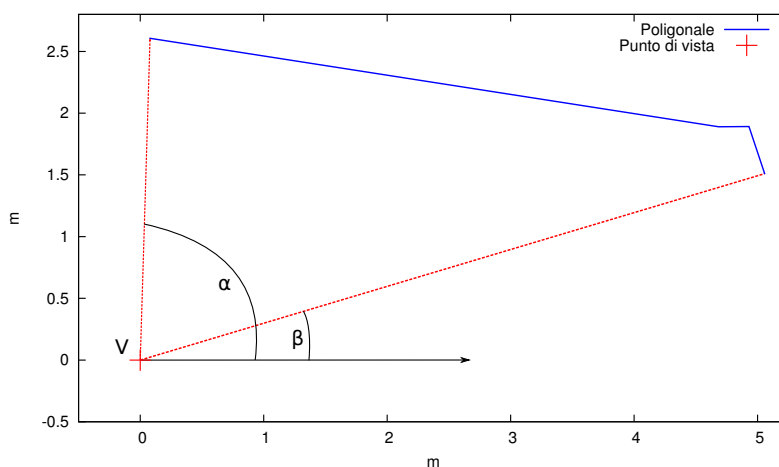


Figura 3.12: Visibilità di una poligonale: gli angoli  $\alpha$  e  $\beta$  rappresentano il massimo e il minimo angolo di visuale con cui è descritta la poligonale.

Oltre alla posizione nell'ordine di visibilità, la distanza reciproca tra due poligonali è un fattore importante per determinare se le poligonali debbano essere unite: infatti l'unione di due poligonali lontane comporterebbe l'inserimento di alcuni segmenti fittizi, che formerebbero un collegamento inesistente, e non giustificato dai dati sensoriali acquisiti. La metrica utilizzata è simile a

quella descritta in 3.3, ma è stata modificata per adattarla alla procedura di unione, in cui deve essere calcolata la distanza minima tra due poligoni. Più precisamente, date due poligoni appartenenti alla mappa,  $P_i$  e  $P_j$ , è necessario calcolare la distanza dei punti più vicini tra di loro, tramite la seguente espressione:

$$D_u(P_i, P_j) = \min \left\{ \min_{r \in P_j} d(P_i, r), \min_{s \in P_i} d(P_j, s) \right\} \quad (3.8)$$

dove la funzione  $d(P_i, r)$  calcola la distanza minima del punto  $r \in P_j$  da tutti i segmenti formati da coppie di punti consecutivi in  $P_i$ . Alcuni esempi di questa distanza sono mostrati in figura 3.13, dove sono calcolate tutte le distanze  $d(P_1, r_j)$ ,  $j = 1, 2, 3$  e  $d(P_2, s_i)$ ,  $i = 1, 2, 3$  per i vertici delle poligoni. Il calcolo di  $D_u(P_1, P_2)$  viene completato scegliendo il minimo tra le distanze  $d(P_1, r_j)$  e tra le distanze  $d(P_2, s_i)$  per le due poligoni, e ancora il minimo tra i due valori ottenuti.

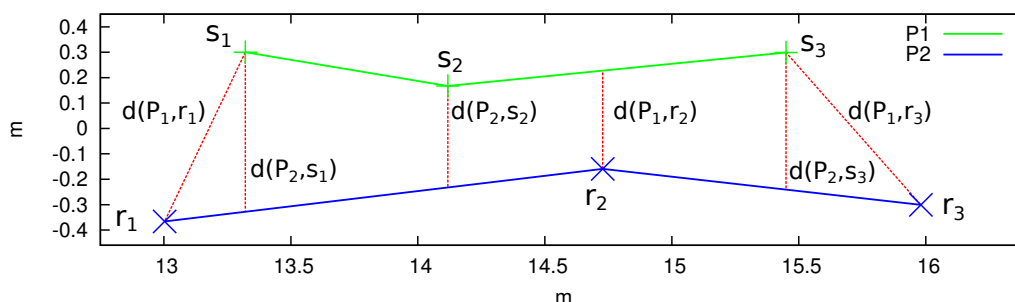


Figura 3.13: Calcolo della distanza  $D_u$ , di equazione 3.8, tra due poligoni.

### 3.5.2 Algoritmo

L'algoritmo 3.2 descrive il riordino dell'insieme di poligoni secondo i criteri di visibilità e distanza esposti in precedenza. Viene utilizzato un vettore di appoggio, *vertici*, che contiene tutti i vertici delle due poligoni. Inizialmente viene individuata la coppia di punti più lontani tra tutti quelli contenuti in *vertici*, e viene scelto quale dei due deve essere inserito per primo nella poligone risultante (*Pout*, righe 1-7). Successivamente per ogni punto *vertici*

l'algoritmo ricerca, tra i segmenti di  $P_{out}$ , formati da due punti consecutivi nella poligonale, quello più vicino al punto  $vertici_i$ . Tra tutte queste coppie punto-segmento viene selezionata quella con distanza reciproca massima (righe 9-15). La distanza punto-segmento usata è la stessa già utilizzata per il calcolo delle metriche in 3.5 e 3.8. La soglia  $S_u$  rappresenta la distanza minima tra punto e segmento oltre la quale la poligonale  $P_{out}$  non viene ulteriormente modificata tramite l'inserimento di punti, e quindi specifica la precisione richiesta nella costruzione della poligonale unita. Nelle righe 17 e 18 la poligonale  $P_{out}$  viene aggiornata con il punto di  $vertici$  selezionato precedentemente, il quale suddivide la poligonale in ulteriori segmenti. Il punto viene poi rimosso dal vettore  $vertici$ , e l'algoritmo termina quando non esistono più punti da inserire in  $P_{out}$  oppure è stata raggiunta la precisione richiesta dalla soglia  $S_u$ .

---

**Algoritmo 3.2** Unione tra due poligoni.

---

**Input:**  $P_1$  prima poligonale;

**Input:**  $P_2$  seconda poligonale;

**Input:**  $V$  punto di vista;

**Input:**  $S_u$  soglia di distanza tra punto e segmento;

**Output:**  $P_{out}$  poligonale risultante;

- 1:  $vertici \leftarrow P_1 + P_2$ , contiene tutti i vertici di  $P_1$  e  $P_2$ ;
- 2:  $p_1, p_2 \leftarrow$  coppia di punti più lontani in  $vertici$ ;
- 3: **if**  $V, p_1, p_2$  sono in disposti in senso antiorario **then**
- 4:      $P_{out} \leftarrow P_{out} + p_1 + p_2$ ;
- 5: **else**
- 6:      $P_{out} \leftarrow P_{out} + p_2 + p_1$ ;
- 7: **end if**
- 8: **while**  $vertici.size > 0$  **and**  $distMax > S_u$  **do**
- 9:     **for**  $i = 0 \rightarrow vertici.size$  **do**
- 10:          $segInd, dist \leftarrow$  indice e distanza del segmento di  $P_{out}$  più vicino al punto  $vertici_i$ ;
- 11:         **if**  $dist > distMax$  **then**
- 12:              $maxInd \leftarrow i$ , indice del punto  $vertici_i$ ;
- 13:              $distMax \leftarrow dist$ , massima distanza tra il punto  $vertici_i$  e il segmento più vicino in  $P_{out}$ ;
- 14:              $segMax \leftarrow segInd$ , indice del segmento di  $P_{out}$  più vicino al punto  $vertici_i$ ;
- 15:         **end if**
- 16:     **end for**
- 17:      $P_{out}^{segMax} \leftarrow vertici_{maxInd}$ , inserimento in ordine in posizione  $segMax$  del punto  $vertici_{maxInd}$ ;
- 18:      $vertici \leftarrow vertici - vertici_{maxInd}$ ;
- 19: **end while**

---

## Capitolo 4

# Localizzazione e setup sperimentale

I sistemi di stoccaggio automatico delle merci in genere operano suddividendo la traiettoria di ogni LGV in segmenti o tratti, tipicamente di lunghezza compresa tra qualche metro e una decina di metri, in cui l'LGV inquadra una medesima area avanzando in essa. È possibile che tale segmentazione sia effettuata già in sede di progettazione e installazione del sistema di LGV e che siano anche costruite le aree di sicurezza dei laser seguendo il layout dell'impianto.

L'LGV transita su questi segmenti di traiettoria durante il suo ciclo di lavoro, e necessita quindi di conoscere in qualunque momento la sua posizione globale nell'impianto. In ambito industriale il metodo di localizzazione più diffuso, e sicuramente il più affidabile, consiste nell'utilizzare dei marker riflettenti disposti nell'ambiente, che possono essere facilmente individuati dal sensore laser tramite le loro proprietà di riflettività. I marker vengono posizionati in modo capillare, per facilitarne l'individuazione, e la loro posizione è nota a priori. L'osservazione di un certo numero di marker da parte dell'LGV, e il confronto con le posizioni dei marker conosciute, permette di effettuare una procedura di triangolazione geometrica per ottenere la posizione globale del robot. Inoltre i sistemi di localizzazione con cui gli LGV sono equipaggiati si

occupano di tenere traccia della posizione del robot nel tempo, in modo che anche in situazioni straordinarie, cioè quando non siano visibili i marker in numero sufficiente, sia possibile ottenere una stima della posa.



Figura 4.1: Fotografia di un impianto industriale in cui si notano un LGV e tre marker riflettenti (immagine tratta da [41]).

Nel caso dell'LGV mostrato in figura 4.1 l'intero procedimento di localizzazione è inglobato nel laser scanner evidenziato, mentre nel contesto applicativo di questa tesi il sensore laser utilizzato non consente di ottenere la posa del robot, non essendo adibito a questo scopo. Conseguentemente, per testare il sistema sviluppato, è stato necessario simulare il comportamento di un LGV sulla piattaforma robotica presente in laboratorio, utilizzando i locali dell'Università degli Studi di Parma come ambiente per i test, nonostante le dimensioni contenute. Per riuscire a simulare anche la fase di localizzazione necessaria al funzionamento del sistema è stato necessario dotare i locali utilizzati di marker riflettenti e sviluppare un procedura di identificazione di questi marker e triangolazione delle loro posizioni per ottenere la posa del robot. Il sistema inoltre è in grado di tenere traccia della posa tramite un filtro di Kalman esteso, come descritto in 4.1.2, proprietà che consente di

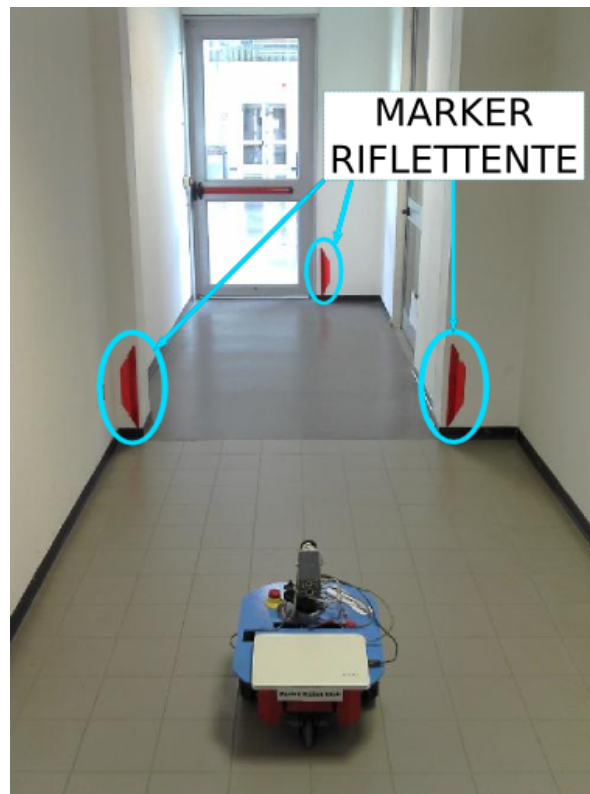


Figura 4.2: Ambiente di navigazione: in evidenza i tre marker riflettenti utilizzati per la localizzazione.

ottenere una stima della posa anche quando non possono essere rilevati i tre marker, mostrati in figura 4.2. Questo può accadere frequentemente nelle simulazioni effettuate con il robot dal momento che i marker sono disposti sul piano di scansione dell'unico sensore disponibile, che è anche utilizzato per il rilevamento degli ostacoli, e quindi i marker stessi possono essere occlusi dagli ostacoli presenti sulla scena, e alcuni di loro possono non essere rilevati. Questo capitolo è suddiviso nelle seguenti sezioni principali:

- *localizzazione*: setup e metodo utilizzato per ricavare la posizione globale del robot partendo dall'insieme di marker disposti nell'ambiente, le cui coordinate rispetto a un sistema di riferimento globale sono note a priori (sezione 4.1) ;

- *pianificazione*: pianificazione della traiettoria del robot tramite curve spline di terzo grado (sezione 4.2) ;
- *generazione e inseguimento*: generazione dei comandi ai motori del robot necessari per l'inseguimento della traiettoria tramite la legge di controllo del sistema (sezione 4.2.1).

## 4.1 Localizzazione

L'obiettivo principali di questa fase consiste nel ottenere la posa del robot (cioè la sua posizione e orientamento) espressa rispetto ad un sistema di riferimento. Nell'applicazione di interesse il sistema di riferimento rispetto al quale avviene la localizzazione può essere specifico per ciascuna mappa locale, oppure assoluto. La semplice odometria del robot può essere utilizzata per costruire una singola mappa locale, ma la procedura di localizzazione qui presentata permette di migliorare la precisione nella stima della posizione, soprattutto nel caso in cui il robot rivisiti di nuovo la mappa locale dopo aver percorso un lungo tratto.

È stato utilizzato un sistema di localizzazione basato su marker artificiali simile a quello utilizzato dagli LGV, e quindi indipendente dalla rappresentazione scelta per le mappe locali. Nella foto 4.2 si possono notare i tre marker utilizzati, che vengono rilevati dal sensore, tramite il controllo delle informazioni di riflettività associate ad ogni beam e il filtraggio dei beam attraverso una soglia sul valore riflesso. Il driver software del laser scanner è stato ampliato per ricavare, tramite interfaccia ethernet, anche questi valori di riflessione per ogni beam, oltre al valore di distanza. I beam interessati da un valore di riflettività alto vengono poi convertiti in coordinate cartesiane rispetto al sistema di riferimento locale del laser, considerando il valore di distanza associato e l'indice del beam nella scansione corrente che indica l'angolo di scansione. Nel caso siano individuati più di tre marker solo i tre più vicini al robot vengono considerati per la triangolazione.

### 4.1.1 Triangolazione

Il sistema di localizzazione impiega un filtro di Kalman esteso, e un metodo di triangolazione utilizzato per inizializzare la posa del filtro. Il problema della triangolazione può essere definito nel modo seguente: note le posizioni di tre marker disposti nell'ambiente  $\mathbf{P}_{L_i} = [x_{L_i}, y_{L_i}]^T$   $i = 1, 2, 3$ , rispetto ad un frame globale, determinare la posizione  $\mathbf{R} = [R_x, R_y]^T$  e l'orientamento  $\mathbf{R}_\theta$  del robot, considerando le osservazioni sensoriali effettuate sui marker e le relative posizioni individuate. Quindi la procedura di risoluzione utilizza le posizioni dei marker misurate manualmente durante la preparazione dell'ambiente di lavoro, e le confronta con l'osservazione attuale in cui sono presenti almeno tre marker, per determinare tre angoli relativi tra l'asse principale del robot e ciascuno dei marker individuati. In figura 4.3 è mostrata una rappresentazione schematica della configurazione robot-marker utilizzata per ricavare un modello matematico che viene risolto tramite l'algoritmo di triangolazione descritto in [40].

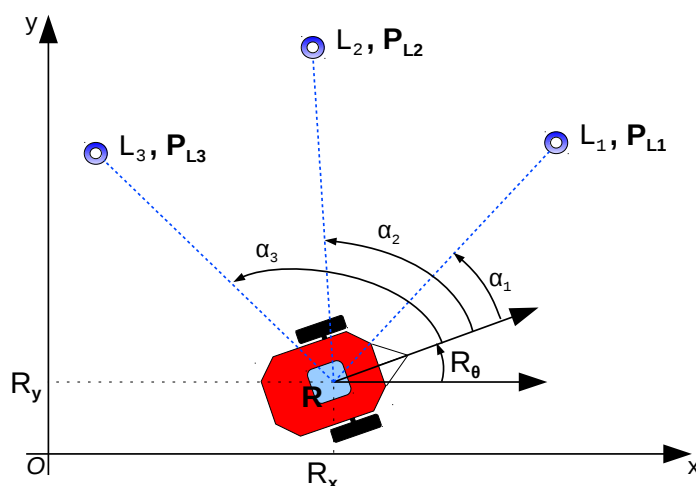


Figura 4.3: Rappresentazione schematica del robot e dei marker da cui ricavare un modello matematico per la soluzione del problema di triangolazione.

L'algoritmo calcola la posizione del robot come intersezione tra due circonferenze che passano da due marker e dalla posizione stessa, evidenziate in figura 4.4. Possono essere definite tre diverse circonferenze, ognuna per ogni

angolo relativo tra l'asse del robot e le linee che congiungono  $\mathbf{R}$  con i marker, cioè gli angoli  $\alpha_{ij} = \alpha_j - \alpha_i$  con  $i, j = 1, 2, 3, i \neq j$ . Nella pratica sono

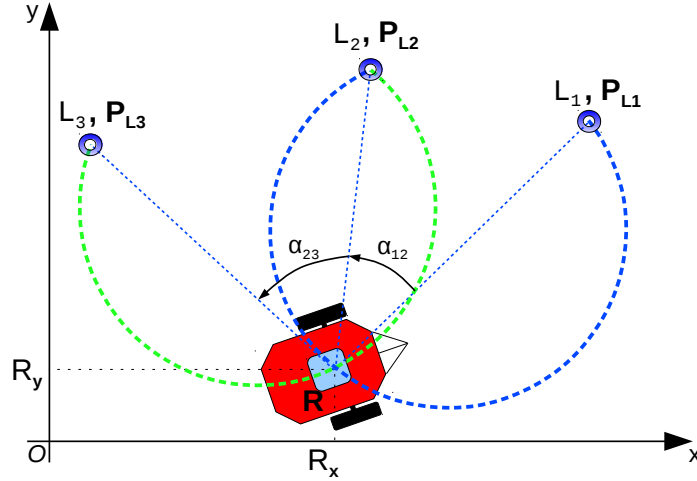


Figura 4.4: Circonferenze passanti per i punti  $L_1, L_2, R$  e  $L_2, L_3, R$ .

sufficienti solo due circonferenze, quelle mostrate in figura 4.4, associate ai due angoli indipendenti  $\alpha_{12}$  e  $\alpha_{23}$ . È possibile scrivere le equazioni delle due circonferenze e risolverne il sistema per ottenere la loro intersezione, cioè la posizione del robot cercata.

Per semplificare i calcoli è stata riportata l'intera costruzione geometrica in un nuovo sistema di riferimento, la cui origine viene posizionata in corrispondenza del marker  $L_2$ , e il sistema è orientato in modo che il marker  $L_3$  risieda sul suo asse  $x$ . Quindi l'angolo di rotazione tra questo sistema e la terna di riferimento globale dello spazio di lavoro, indicata con  $O$  in figura 4.4, risulta essere

$$\varphi = \arctan \frac{y_{L3} - y_{L2}}{x_{L3} - x_{L2}} \quad (4.1)$$

dove  $[x_{Li}, y_{Li}]$  sono le coordinate globali del marker  $i$  rispetto alla terna  $O$ . La matrice di rotazione  $\mathbf{S}_\varphi$ , ottenuta dall'angolo  $\varphi$ , consente di calcolare la posizione dei marker nel nuovo sistema di riferimento

$$\bar{\mathbf{P}}_{Li} = \mathbf{S}_\varphi^{-1} (\mathbf{P}_{Li} - \mathbf{P}_{L2}) \quad (4.2)$$

Quindi, dato il nuovo sistema di coordinate scelto, le posizioni dei marker in 4.2 sono date da  $\bar{\mathbf{P}}_{L1} = [\bar{x}_{L1}, \bar{y}_{L1}]^T$ ,  $\bar{\mathbf{P}}_{L2} = [0, 0]^T$  e  $\bar{\mathbf{P}}_{L3} = [\bar{x}_{L3}, 0]^T$ . Riscrivendo le equazioni delle circonferenze tramite queste nuove coordinate è possibile risolvere il sistema di equazioni e ricavare il vettore posizione del robot  $\bar{\mathbf{R}}$  nelle nuove coordinate. Questo vettore può essere riportato nelle coordinate iniziali tramite la trasformazione

$$\mathbf{R} = [x_{L2}, y_{L2}]^T + \mathbf{S}_\varphi \bar{\mathbf{R}} \quad (4.3)$$

Successivamente, una volta nota la posizione del robot  $\mathbf{R}$ , l'orientazione rispetto alla terna di riferimento globale  $O$  si ottiene con

$$\mathbf{R}_\theta = \arctan \frac{y_{Li} - R_y}{x_{Li} - R_x} - \alpha_i \quad (4.4)$$

dove  $[x_{Li}, y_{Li}]$  sono le coordinate del marker  $i$  e  $\alpha_i$  è il corrispondente angolo. Il posizionamento dei marker influisce sull'errore commesso dalla procedura di triangolazione, in particolare nel caso in cui i tre marker siano disposti su una unica circonferenza e il robot si trovi su questa circonferenza, l'errore di posizione è massimo, ed è derivato da una singolarità comune a tutti gli algoritmi di triangolazione geometrica, come già descritto in 1.1.1. Nei test effettuati i marker sono stati disposti a distanze sufficienti e in posizioni tali da evitare questo problema.

### 4.1.2 Localizzatore con filtro di Kalman

Il procedimento descritto in precedenza necessita di almeno tre marker visibili nella scansione corrente per localizzare il robot. In molte situazioni non è possibile osservare tutti i marker presenti, principalmente a causa di occlusione e dei limiti sul raggio di azione dei sensori. In questi casi è quindi necessario adottare una tecnica di inseguimento della posa del robot per disporre di una stima affidabile durante tutta la percorrenza della traiettoria. La tecnica di tracciamento adottata si basa sull'utilizzo di un filtro di Kalman esteso che viene inizializzato con la posa globale, ottenuta mediante la

triangolazione descritta precedentemente, a robot fermo. Questa inizializzazione è l'unica fase in cui è necessario osservare tutti i tre marker, per poter effettuare la triangolazione. Il vettore di stato del sistema, in seguito indicato con  $\mathbf{p}_t$ , è composto dalla posa del robot, mentre il modello del moto utilizzato consiste nell'aggiornamento dello stato al tempo  $t - 1$  tramite composizione con il valore differenziale di odometria accumulato tra il tempo  $t$  e l'istante precedente  $t - 1$ . Denominato  $u_t$  questo valore di odometria a cui corrisponde un ingresso del sistema, il modello è descritto dall'equazione 4.5.

$$\mathbf{p}_t = g(\mathbf{p}_{t-1}, \mathbf{u}_t) = \mathbf{p}_{t-1} \oplus \mathbf{u}_t = \begin{bmatrix} p_t^x + u_t^x \cos p_t^\theta - u_t^y \sin p_t^\theta \\ p_t^y + u_t^x \sin p_t^\theta + u_t^y \cos p_t^\theta \\ p_t^\theta + u_t^\theta \end{bmatrix} \quad (4.5)$$

Il modello sensoriale utilizzato per descrivere le osservazioni effettuate tramite laser scanner è mostrato in 4.6.

$$\mathbf{z}_t = h(\mathbf{p}_t, \mathbf{l}) = \mathbf{l} \ominus \mathbf{p}_t = \begin{bmatrix} (l^x - p_t^x) \cos p_t^\theta + (l^y - p_t^y) \sin p_t^\theta \\ -(l^x - p_t^x) \sin p_t^\theta + (l^y - p_t^y) \cos p_t^\theta \end{bmatrix} \quad (4.6)$$

dove  $\ominus$  rappresenta l'operazione di decomposizione, e  $\mathbf{l} = [l^x, l^y]$  rappresenta la posizione globale di una marker ottenuta attraverso la misurazione manuale delle coordinate sul campo di lavoro del robot. I marker osservati durante il movimento del robot devono essere associati ai marker di cui si conosce la posizione, in modo che il filtro sia in grado di calcolare una stima sulle osservazioni correnti ( $\mathbf{z}_t$ ), che viene utilizzata per la correzione dello stato predetto, come si può vedere nell'equazione 4.12. La procedura di associazione determina per ogni marker osservato qual'è il marker conosciuto più vicino. Nel caso venga identificato un marker in una posizione non nota questo non viene utilizzato dal filtro per la correzione, siccome non è possibile associarlo ad un marker conosciuto. Per ottenere il valore di correzione dello stato al tempo  $t$ , mediante la fase di correzione del filtro, è necessario linearizzare le equazioni del modello del moto e del modello sensoriale, e calcolarle nei valori dello stato ritenuti più probabili nell'istante corrente. Il modello di moto  $g$  viene linearizzato considerando il valore medio  $\boldsymbol{\mu}$  dello stato corretto

al tempo  $t - 1$ . L'espressione dello jacobiano in  $\boldsymbol{\mu}_{t-1}$  è mostrata in 4.7. Per quanto riguarda il modello sensoriale il punto di linearizzazione ritenuto più affidabile è il valore medio dello stato predetto al tempo  $t$ . La linearizzazione avviene quindi attorno a  $\bar{\boldsymbol{\mu}}_t$ , e lo jacobiano corrispondente è mostrato in 4.8.

$$\mathbf{G}_t = \frac{\partial g}{\partial \mathbf{p}}(\boldsymbol{\mu}_{t-1}, \mathbf{u}) = \begin{bmatrix} 1 & 0 & -u_t^x \sin \mu_{t-1}^\theta - u_t^y \cos \mu_{t-1}^\theta \\ 0 & 1 & u_t^x \cos \mu_{t-1}^\theta - u_t^y \sin \mu_{t-1}^\theta \\ 0 & 0 & 1 \end{bmatrix} \quad (4.7)$$

$$\begin{aligned} \mathbf{H}_t &= \frac{\partial h}{\partial \mathbf{p}}(\bar{\boldsymbol{\mu}}_t, \mathbf{l}) \\ &= \begin{bmatrix} -\cos \bar{\mu}_t^\theta & -\sin \bar{\mu}_t^\theta & -(l^x - \bar{\mu}_t^x) \sin \bar{\mu}_t^\theta + (l^y - \bar{\mu}_t^y) \cos \bar{\mu}_t^\theta \\ \sin \bar{\mu}_t^\theta & -\cos \bar{\mu}_t^\theta & -(l^x - \bar{\mu}_t^x) \cos \bar{\mu}_t^\theta - (l^y - \bar{\mu}_t^y) \sin \bar{\mu}_t^\theta \end{bmatrix} \end{aligned} \quad (4.8)$$

Le equazioni del filtro di Kalman esteso sono riportate nel seguito per completezza: le equazioni 4.9 e 4.10 mostrano il calcolo dello stato predetto e della sua matrice di covarianza.

$$\bar{\boldsymbol{\mu}}_t = g(\boldsymbol{\mu}_{t-1}, \mathbf{u}_t) \quad (4.9)$$

$$\bar{\boldsymbol{\Sigma}}_t = \mathbf{G}_t \boldsymbol{\Sigma}_{t-1} \mathbf{G}_t^T + \mathbf{R}_t \quad (4.10)$$

L'equazione 4.11 mostra il calcolo del guadagno del filtro di Kalman, utilizzato successivamente per la fase di correzione dello stato, 4.12, e per l'aggiornamento della covarianza, 4.13.

$$\mathbf{K}_t = \bar{\boldsymbol{\Sigma}}_t \mathbf{H}_t^T (\mathbf{H}_t \bar{\boldsymbol{\Sigma}}_t \mathbf{H}_t^T + \mathbf{Q}_t)^{-1} \quad (4.11)$$

$$\boldsymbol{\mu}_t = \bar{\boldsymbol{\mu}}_t + \mathbf{K}_t (\mathbf{z}_t - h(\bar{\boldsymbol{\mu}}_t, \mathbf{l})) \quad (4.12)$$

$$\boldsymbol{\Sigma}_t = (\mathbf{I} - \mathbf{K}_t \mathbf{H}_t) + \bar{\boldsymbol{\Sigma}}_t \quad (4.13)$$

$\mathbf{R}_t$  e  $\mathbf{Q}_t$  rappresentano le matrici di covarianza associate rispettivamente all'incertezza sulla posa del robot e sull'osservazione sensoriale.

Al fine di stimare la stabilità del filtro l'entità della correzione operata dal filtro è stata confrontata con la relativa deviazione standard per ciascuna delle variabili di stato. La figura 4.5 mostra l'andamento dei valori durante le iterazioni del filtro. Durante ogni iterazione il filtro viene aggiornato con

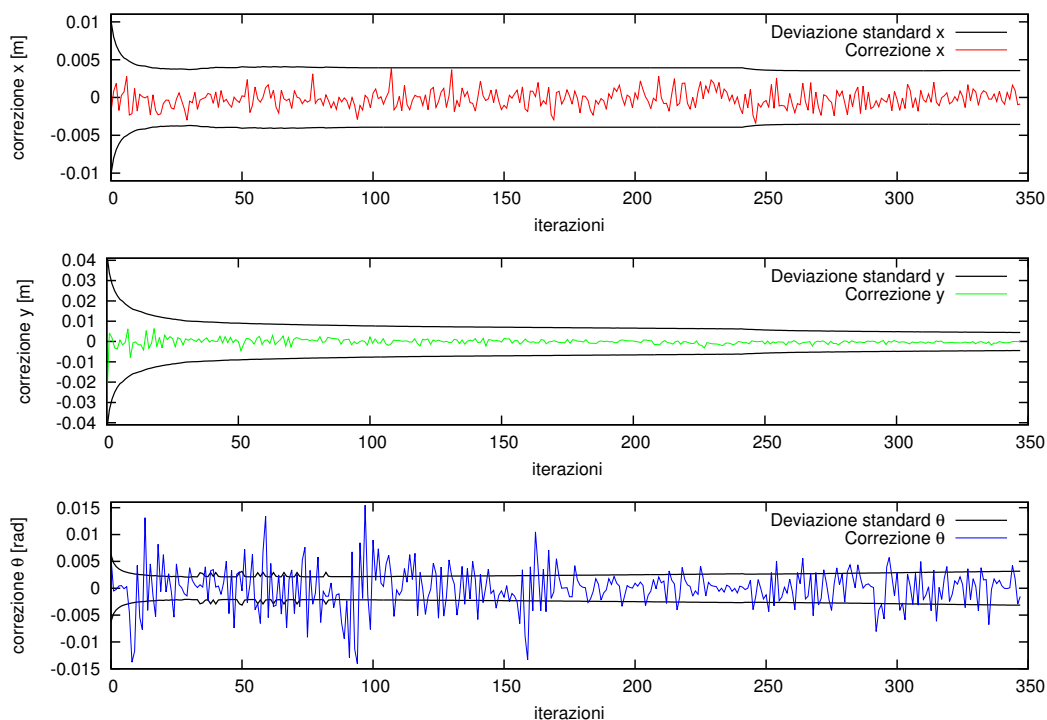


Figura 4.5: Valori delle correzioni operate dal filtro di Kalman durante un prova sperimentale, lungo le direzioni  $x$ ,  $y$  e sull'orientazione  $\theta$  del robot.

un nuovo valore di odometria per la fase di predizione, e con l'insieme dei marker correntemente osservati, per eseguire la correzione dello stato. Ogni iterazione dura in media  $40\text{ ms}$ .

Per quanto riguarda l'esperimento in figura 4.5 si osserva la relativa stabilità della stima delle variabili di stato di posizione ed un maggiore difficoltà nella stima dell'orientamento, che è la componente più difficile da stimare.

## 4.2 Inseguimento di un percorso pianificato

Il sistema sviluppato è in grado di localizzare correttamente il robot, e di tenere traccia della sua posa anche in situazioni particolari, anche quando sono osservati pochi marker. È quindi possibile pianificare il movimento del robot, attraverso l'utilizzo di una traiettoria descritta dalle equazioni di una curva geometrica. La traiettoria viene definita come l'unione di curve spline ciascuna corrispondente ad un segmento di traiettoria. Il robot è in grado di inseguire la traiettoria pianificata utilizzando la posa globale individuata dalla fasi precedenti.

Questa sezione descrive la pianificazione e generazione di un segmento di traiettoria tramite curve spline di terzo grado. Definito un parametro  $s \in [0, 1]$ , il punto della spline corrispondente verrà indicato come  $p(s) = [x(s), y(s)]^T$ , e la derivata prima della traiettoria rispetto a  $s$  sarà indicata con  $p'(s) = [x'(s), y'(s)]^T$ . I valori di  $x(s), y(s)$  e  $x'(s), y'(s)$  sono dati rispettivamente dalle equazioni della spline di terzo grado e dalle loro derivate prime

$$x(s) = x_0 + x_1 s + x_2 s^2 + x_3 s^3 \quad (4.14)$$

$$y(s) = y_0 + y_1 s + y_2 s^2 + y_3 s^3 \quad (4.15)$$

$$x'(s) = x_1 + 2x_2 s + 3x_3 s^2 \quad (4.16)$$

$$y'(s) = y_1 + 2y_2 s + 3y_3 s^2 \quad (4.17)$$

La pianificazione della traiettoria deve rispettare i vincoli geometrici descritti dal punto iniziale e dal punto finale scelti. Definito  $p'(s) = [x'(s), y'(s)]^T$ , questi vincoli risultano essere:

- la traiettoria parte in  $P_A = [x_A, y_A]^T$ , con orientazione iniziale  $\theta_A$ ;
- la traiettoria finisce in  $P_B = [x_B, y_B]^T$ , con orientazione finale  $\theta_B$ ;
- la velocità iniziale è data da:

$$p'(0) = \eta_A \begin{bmatrix} \cos \theta_A \\ \sin \theta_A \end{bmatrix} \quad (4.18)$$

- la velocità finale è data da:

$$p'(1) = \eta_B \begin{bmatrix} \cos \theta_B \\ \sin \theta_B \end{bmatrix} \quad (4.19)$$

I parametri  $\eta_A$  e  $\eta_B$  influenzano la forma della curva nel tratto iniziale e finale, e possono avere valori arbitrari. In questo sistema è stata utilizzata la lunghezza della curva come valore per entrambi i parametri. Imponendo le condizioni appena descritte vengono ricavate le equazioni dei coefficienti della spline:

$$x_0 = x_A \quad (4.20)$$

$$x_1 = \eta_A \cos \theta_A \quad (4.21)$$

$$x_2 = 3(x_B - x_A) - 2\eta_A \cos \theta_A - \eta_B \cos \theta_B \quad (4.22)$$

$$x_3 = -2(x_B - x_A) + \eta_A \cos \theta_A + \eta_B \cos \theta_B \quad (4.23)$$

$$y_0 = y_A \quad (4.24)$$

$$y_1 = \eta_A \sin \theta_A \quad (4.25)$$

$$y_2 = 3(y_B - y_A) - 2\eta_A \sin \theta_A - \eta_B \sin \theta_B \quad (4.26)$$

$$y_3 = -2(y_B - y_A) + \eta_A \sin \theta_A + \eta_B \sin \theta_B \quad (4.27)$$

La pianificazione a tratti di segmenti di traiettoria non rettilinei consente di ottenere un comportamento più realistico del robot, rispetto all'utilizzo delle spline in un unico segmento pianificato, che porterebbe ad una difficoltà nell'ottenere movimenti complessi e articolati del robot. Data una sequenza di punti da visitare ( $P_0, \dots, P_n$ ) ogni segmento  $i = 1, \dots, n$  della traiettoria viene pianificato in modo indipendente, considerando come punto iniziale  $P_{i-1}$  e come punto finale  $P_i$ . I comandi per i motori del robot sono ricavati dalla fase di generazione dei controlli descritta in 4.2.1.

### 4.2.1 Generazione dei controlli

La piattaforma robotica utilizzata per eseguire i tesse del sistema è una Pioneer 3-DX [43], mostrata in figura 4.6. Si tratta di un robot uniciclo, composto da due ruote motrici anteriori e una ruota caster posteriore. Il mo-



Figura 4.6: Piattaforma robotica Pioneer P3DX, utilizzata per i test.

dello cinematico associato a questo tipo di robot è composta dalle seguenti equazioni

$$\begin{cases} \dot{x}_r = u_{1,r} \cos \theta_r \\ \dot{y}_r = u_{1,r} \sin \theta_r \\ \dot{\theta}_r = u_{2,r} \end{cases} \quad (4.28)$$

dove  $[x_r, y_r]^T$  rappresenta la posizione del punto medio dell'asse delle ruote motrici,  $\theta_r$  l'orientamento del robot rispetto ad un frame globale, mentre  $u_{1,r}$  e  $u_{2,r}$  sono rispettivamente la velocità lineare e angolare del veicolo.

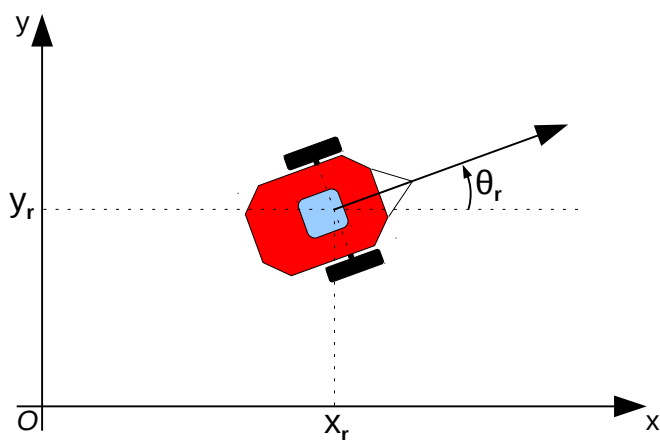
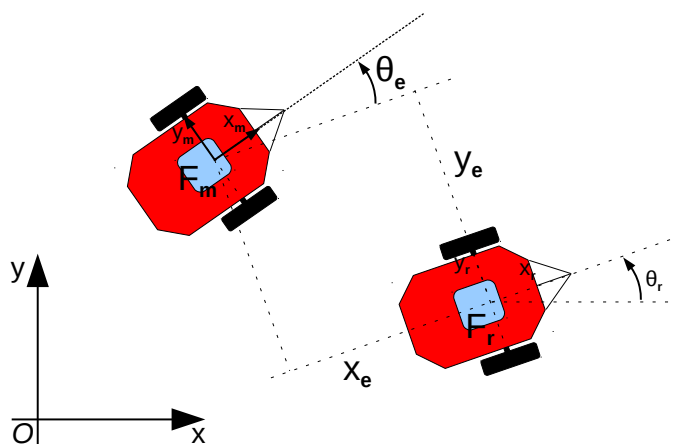


Figura 4.7: Modello del robot uniciclo.

La figura 4.7 mostra il modello utilizzato. Il metodo di controllo basato sulla *chain form*, descritta in [45], prevede l'utilizzo di un robot di riferimento, a cui è associato rigidamente il frame  $F_r$ , in figura 4.8, e la presenza di un robot reale, rappresentato da  $F_m$ , da controllare.

Figura 4.8: Robot di riferimento  $F_r$ , e robot da controllare  $F_m$ .

Il problema consiste nel tracciare la posizione e l'orientazione del robot di riferimento che percorre la traiettoria pianificata e progettare il controllo in modo che stabilizzi asintoticamente la traiettoria del robot reale sulla traiettoria di riferimento; questo significa ridurre asintoticamente a zero l'errore di

posizione  $[x_m - x_r, y_m - y_r, \theta_m - \theta_r]$ . Per ottenere la soluzione cercata è necessario che la traiettoria di riferimento sia fattibile, e in particolare questa deve essere una funzione  $C^1$  (o  $C^1$  a tratti), soluzione del modello cinematico del robot quando è dato uno specifico input,  $u_r(t)$ , chiamato appunto controllo di riferimento.

La parte dell'errore di tracciamento relativa alla posizione, cioè  $[x_e, y_e]$ , si può ricavare dalla figura 4.8, e può essere definita come segue

$$\begin{bmatrix} x_e \\ y_e \end{bmatrix} = \begin{bmatrix} \cos \theta_r & \sin \theta_r \\ -\sin \theta_r & \cos \theta_r \end{bmatrix} \begin{bmatrix} x_m - x_r \\ y_m - y_r \end{bmatrix} \quad (4.29)$$

La derivata prima rispetto al tempo dell'errore di posizione risulta essere

$$\begin{bmatrix} \dot{x}_e \\ \dot{y}_e \end{bmatrix} = \dot{\theta}_r \begin{bmatrix} -\sin \theta_r & \cos \theta_r \\ -\cos \theta_r & -\sin \theta_r \end{bmatrix} \begin{bmatrix} x_m - x_r \\ y_m - y_r \end{bmatrix} \quad (4.30)$$

$$+ \begin{bmatrix} \cos \theta_r & \sin \theta_r \\ -\sin \theta_r & \cos \theta_r \end{bmatrix} \begin{bmatrix} \dot{x}_m - \dot{x}_r \\ \dot{y}_m - \dot{y}_r \end{bmatrix} \quad (4.31)$$

$$= \begin{bmatrix} u_{2,r} y_e + u_1 \cos(\theta_m - \theta_r) - u_{1,r} \\ -u_{2,r} x_e + u_1 \sin(\theta_m - \theta_r) \end{bmatrix} \quad (4.32)$$

L'errore di orientazione,  $\theta_e = \theta_m - \theta_r$ , completa le equazioni del sistema non lineare associato all'errore di tracciamento fra le terne  $F_m$  e  $F_r$ . Considerando che  $\dot{\theta}_e = \dot{\theta}_m - \dot{\theta}_r = u_2 - u_{2,r}$  si ha

$$\begin{cases} \dot{x}_e = u_{2,r} y_e + u_1 \cos \theta_e - u_{1,r} \\ \dot{y}_e = -u_{2,r} x_e + u_1 \sin \theta_e \\ \dot{\theta}_e = u_2 - u_{2,r} \end{cases} \quad (4.33)$$

Per ricavare le variabili di controllo  $[u_1, u_2]$ , il sistema 4.33 viene trasformato nella chain form tramite il seguente cambio di variabili

$$z_1 = x_e \quad (4.34)$$

$$z_2 = y_e \quad (4.35)$$

$$z_3 = \tan \theta_e \quad (4.36)$$

$$w_1 = u_1 \cos \theta_e - u_{1,r} \quad (4.37)$$

$$w_2 = \frac{u_2 - u_{2,r}}{\cos^2 \theta_e} \quad (4.38)$$

Nelle nuove variabili il sistema in 4.33 risulta

$$\begin{cases} \dot{z}_1 = u_{2,r} z_2 + w_1 \\ \dot{z}_2 = -u_{2,r} z_1 + u_{1,r} z_3 + w_1 z_3 \\ \dot{z}_3 = w_2 \end{cases} \quad (4.39)$$

Nell'ambito di quest'ultimo sistema è possibile individuare una legge di controllo che rende il punto di equilibrio nell'origine stabile globalmente e asintoticamente. Questo implica che il segnale errore tende asintoticamente a zero e il robot è in grado di uniformare il suo percorso con la traiettoria di riferimento. La legge di controllo risulta essere

$$\begin{cases} w_1 = -K_1 |u_{1,r}| (z_1 + z_2 z_3) \\ w_2 = -K_2 u_{1,r} z_2 - K_3 |u_{1,r}| z_3 \end{cases} \quad (4.40)$$

Questa legge può essere semplificata linearizzando il sistema 4.40 rispetto alla variabile  $z$  attorno all'origine, dove, per piccoli valori di  $\theta_e$ , si ha che  $w_1 \approx u_1 - u_{1,r}$  e  $w_2 \approx u_2 - u_{2,r}$ . Il risultato è mostrato in 4.41, dove sono stati inseriti i valori delle variabili  $z_1$ ,  $z_2$  e  $z_3$  per ottenere la legge di controllo utilizzata in questo lavoro.

$$\begin{cases} u_1 = u_{1,r} - K_1 |u_{1,r}| x_e \\ u_2 = u_{2,r} - K_2 u_{1,r} y_e - K_3 |u_{1,r}| \tan \theta_e \end{cases} \quad (4.41)$$

Le costanti  $K_1$ ,  $K_2$  e  $K_3$  possono essere scelte in modo da ottenere un sistema asintoticamente stabile. Questo controllo stabilizza globalmente e asintoticamente il sistema quando  $u_{1,r}$  e  $u_{2,r}$  sono costanti e  $u_{1,r} \neq 0$ . Inoltre il controllo è stato progettato in modo che i valori ottimi delle costanti  $K$  quando le velocità sono  $u_{1,r} = 1$  e  $u_{2,r} = 0$  diano buoni risultati anche per tutte le altre velocità. Quindi la legge di controllo utilizzata nei test si ottiene imponendo  $u_r = [1, 0]^T$  e  $K_1 = 3$ ,  $K_2 = 1.5$ ,  $K_3 = 1.5$ .

# Capitolo 5

## Risultati

In questo capitolo vengono descritti alcuni esperimenti effettuati nei locali del Dipartimento di Ingegneria dell'Informazione dell'Università degli Studi di Parma, al fine di validare e stimare la precisione e l'affidabilità del metodo proposto nella costruzione della mappa, ed in particolare nell'individuazione di oggetti quasi-statici. Nel seguito vengono esposte le modalità di esecuzione dei test, e i risultati ottenuti.

Le prove sono consistite nel transito ripetuto del robot mobile in ambienti specifici, adatti a simulare il passaggio dell'LGV in un segmento di traiettoria. Lungo il percorso sono stati posizionati marker artificiali ed ostacoli, come si può notare in figura 5.1. Le aree di sicurezza del robot sono state scelte in modo da limitare la loro intersezione con le pareti e gli altri elementi statici della scena, senza penalizzare troppo la rilevazione di oggetti durante il movimento, e sono descritte nell'appendice A. Gli oggetti presenti nella scena non ricadono nella zona di transito fisico del veicolo, mentre, per semplificare e velocizzare la procedura di test, il robot non si arresta quando un ostacolo ricade nell'area di sicurezza più interna. Gli scenari utilizzati per i test sono denominati Pal1 (in Figura 5.1), in cui è stato eseguito un solo test, e Pal3 (in Figura 5.6), dove sono state effettuate due prove denominate nel seguito Pal3a e Pal3b. Altri test sono stati realizzati tramite un simulatore di piattaforme robotiche; gli scenari interessati verranno denominati



Figura 5.1: Ambiente in cui è stato eseguito uno dei test: si notano i tre landmark riflettenti e alcuni ostacoli statici.

Sim1 e Sim2. In queste prove sperimentali è stata valutata l'accuratezza dell'algoritmo di costruzione della mappa. Pertanto sono stati introdotti nella scena solo oggetti statici o quasi-statici; la presenza di oggetti dinamici sarà discussa nella sezione 5.3.1.

Ciascun test si è svolto nel modo seguente: al primo transito del robot sulla scena il sistema percepisce e archivia gli oggetti quasi-statici e statici presenti nelle aree di guardia del laser scanner, e costruisce la mappa locale di poligonali che verrà poi impiegata come mappa di riferimento per le prove successive. In seguito il robot transita altre quattro volte nella stessa scena immutata acquisendo mediamente 350 scansioni ad ogni transito nell'ambiente Pal1 e 500 nell'ambiente Pal3 e in simulazione. Queste scansioni, o viste, acquisite durante il moto del robot in un segmento di traiettoria,

vengono fuse in un'unica rappresentazione dello stato attuale del segmento, la quale viene a sua volta arricchita durante il transito con i nuovi oggetti percepiti. Nel primo attraversamento, al segmento di traiettoria è associato l'insieme di poligoni ottenuto da un numero elevato di scansioni, tutte riferite rispetto alla prima posa del segmento stesso in cui si trova il robot. Nel secondo attraversamento, così come in tutti i successivi, per ogni segmento è disponibile una rappresentazione di ciò che era stato osservato in precedenza. È possibile quindi operare un confronto della rappresentazione corrente con quella di riferimento, per determinare i cambiamenti avvenuti, oppure per verificare che le poligoni osservate precedentemente siano ancora osservabili e abbiano le stesse caratteristiche. Questa analisi dei dati sensoriali può essere interessata da molteplici problemi: ad esempio, due oggetti inseriti nella rappresentazione del transito precedente possono essere rilevati come uno solo nel nuovo transito; oppure un oggetto rilevato in precedenza può essere visto come più oggetti nel nuovo transito. Gli algoritmi adottati cercano di ridurre questi fenomeni dovuti a rumore ed incertezze di misura; infatti, associazioni non corrette dei dati possono portare al mancato riconoscimento di un oggetto come quasi-statico, e quindi ad un comportamento errato o non sicuro da parte dell'LGV. In generale si possono verificare le seguenti situazioni durante l'esplorazione del robot:

- vengono rilevate poligoni in un'area in cui erano assenti in precedenza: si tratta di nuovi dati sensoriali ancora da classificare, che devono essere integrati nella mappa;
- alcune delle poligoni precedentemente rilevate ora risultano assenti: in questo caso le poligoni potrebbero essere associate a oggetti dinamici o quasi-statici che non sono più presenti nella scena, e quindi la rappresentazione deve essere aggiornata di conseguenza;
- alcune delle poligoni già presenti vengono nuovamente rilevate con le stesse caratteristiche: l'oggetto che genera la rilevazione viene classificato come quasi-statico;

- non sono presenti poligoni nella scansione corrente che ricadono nelle aree di sicurezza, quindi il sistema aggiorna solo la posa del robot.

Un caso di particolare interesse è la presenza di persone o oggetti dinamici che si muovono nell'area di sicurezza. Per verificare l'affidabilità del sistema anche in queste condizioni sono stati effettuati alcuni test in presenza di persone in movimento, e i risultati sono mostrati in 5.3.1, insieme alle problematiche individuate.

L'obiettivo degli esperimenti è una stima approssimata dell'errore commesso nell'identificazione degli elementi della scena, nei passaggi successivi al primo. In particolare vengono calcolate le distanze medie tra oggetti contenuti nella mappa e oggetti presenti nella scansione ottenuta nel transito corrente. Un esempio di questo confronto viene mostrato in figura 5.2, come caso particolare di calcolo della distanza tra due poligoni associate, una appartenente alla scansione, e l'altra inclusa nella mappa di riferimento.

## 5.1 Distanza tra poligoni

Il confronto tra poligoni, che determina una stima dell'errore di identificazione degli oggetti, necessita di una distanza, o metrica, per essere quantificato. La distanza utilizzata in questi test è molto simile a quella già presentata in 3.3 e ne eredita anche le principali proprietà di invarianza e generalità. Tuttavia la distanza di Hausdorff modificata risulta peggiorativa poiché un errore elevato può essere riscontrato quando si confronta una poligonale ottenuta da una vista parziale dell'ostacolo con quella ottenuta da una ricostruzione più completa. Per cercare di attenuare questi effetti peggiorativi è stata ridefinita la distanza come segue. Siano  $P$  e  $Q$  due poligoni aventi rispettivi vertici  $[P_0, \dots, P_n]$  e  $[Q_0, \dots, Q_m]$ . Si definiscono gli indici

$$idq_i = \min_{j=1, \dots, m} dist(P_i, Q_{j-1}Q_j) \quad (5.1)$$

dove  $dist(\cdot, \cdot)$  è la distanza punto-segmento: quindi  $idq_i$  rappresenta l'indice del segmento di  $Q$  più vicino al vertice  $P_i$ . Dati  $idq_{min} = \min\{idq_0, \dots, idq_n\}$

e  $idq_{max} = \max\{idq_0, \dots, idq_n\}$  si definisce  $Q|_P$ , poligonale  $Q$  ristretta a  $P$ , la poligonale formata dai vertici  $[Q_{idq_{min}}, Q_{idq_{min}+1}, \dots, Q_{idq_{max}}]$ . La distanza di Hausdorff modificata viene applicata alle due poligonali ristrette, cioè  $Q|_P$  e  $P|_Q$ .

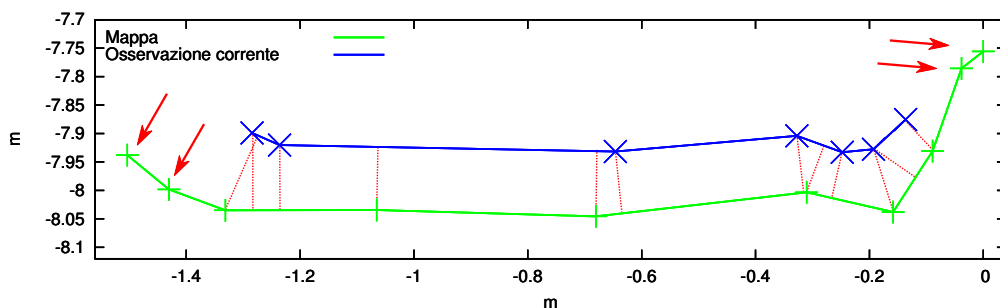


Figura 5.2: Esempio di calcolo della distanza tra poligonali ottenute in due diversi passaggi dello stesso test. La distanza viene accumulata per ogni poligonale della mappa, e vengono calcolate media e deviazione standard dopo l'ultimo passaggio del test.

In figura 5.2 è mostrato come esempio il confronto tra le due poligonali: solamente i punti collegati dai segmenti tratteggiati vengono considerati per il calcolo della distanza. I punti segnalati in rosso nella mappa non intervengono sul risultato finale del calcolo della distanza, non essendo presenti punti altrettanto vicini nella poligonale corrente. Questo consente di ottenere stime più realistiche nella comparazione di poligonali, rispetto all'applicazione della distanza 3.3.

## 5.2 Risultati delle simulazioni

Le prove in simulazione sono caratterizzate da un ridotto errore sull'odometria del robot. Non essendo disponibili le informazioni relative alla riflettività del sensore laser non è possibile individuare i marker, e di conseguenza non è possibile impiegare la procedura di localizzazione. Quindi le scansioni sono state allineate con i soli dati dell'odometria.

I test si sono svolti in due mappe realizzate appositamente, che sono raffigurate sopra alle mappe di riferimento create dall'applicazione durante il primo passaggio del robot. Per questo tipo di test sono stati effettuati tre passaggi del robot per ogni mappa, compreso il primo.

In figura 5.3 è presentata la mappa relativa allo scenario Sim1, con estensione di circa  $40 m^2$ , in cui il robot ha percorso circa 8 metri. La tabella 5.1 riporta

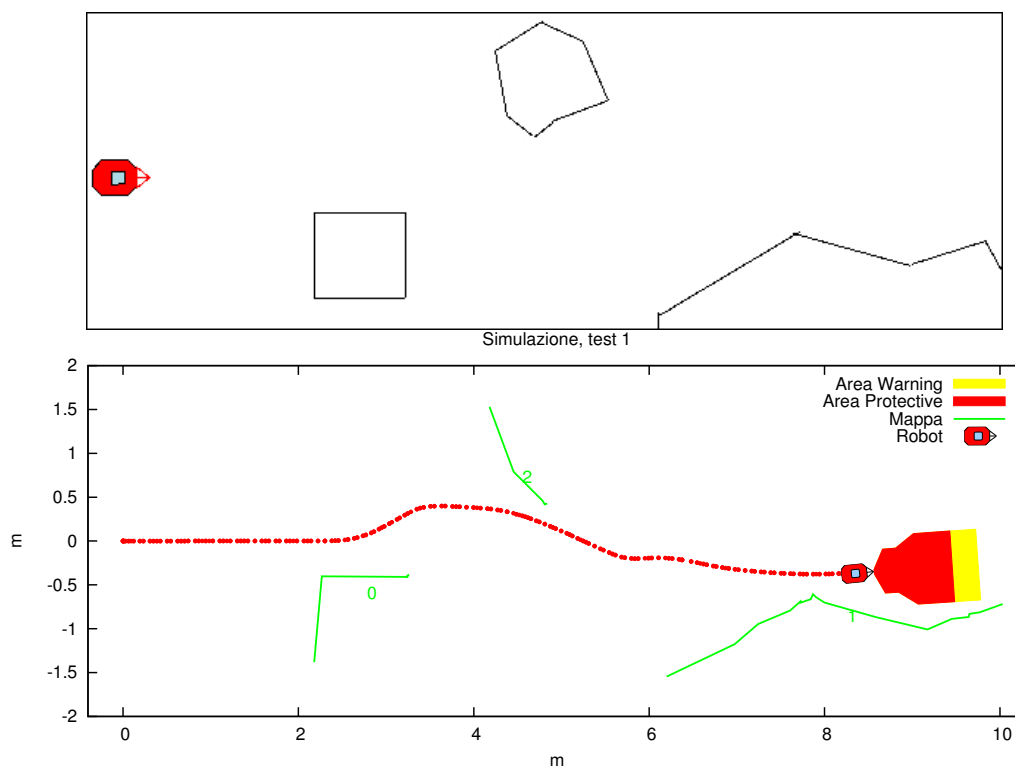


Figura 5.3: Mappa ottenuta in simulazione, utilizzando la mappa denominata Sim1, raffigurata in alto.

i valori medi e le deviazioni standard delle distanze di associazione delle poligoni individuate nel primo passaggio del robot, chiamate  $M_0, \dots, M_n$ , e immagazzinate nella mappa di riferimento. Si può notare come la distanza tra le poligoni associate durante la fase di test rimane contenuta anche con poligoni lunghe e di forma complessa.

In figura 5.4 è mostrata la mappa relativa allo scenario Sim2, con estensione di circa  $63 m^2$ , in cui la traiettoria ha una lunghezza di circa 14 metri.

	$M_0$	$M_1$	$M_2$
M [cm]	2.19	6.62	5.32
DS [cm]	1.73	2.69	1.25

Tabella 5.1: Distanze medie tra le poligonali della mappa ( $M_0, \dots, M_2$ ) e quelle osservate nei passaggi successivi, nel primo test in simulazione (Sim1).

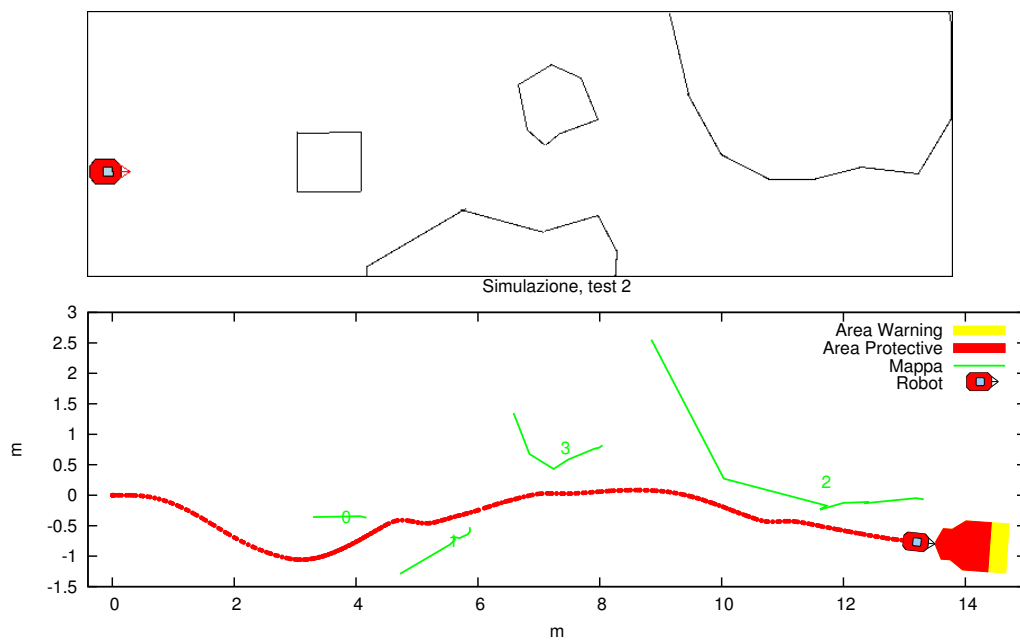


Figura 5.4: Mappa ottenuta in simulazione, utilizzando la mappa denominata Sim2, raffigurata in alto.

La tabella 5.2 riporta i valori ottenuti per questo secondo test. Il valore alto

	$M_0$	$M_1$	$M_2$	$M_3$
M [cm]	1.20	1.63	12.37	2.63
DS [cm]	0.91	1.05	5.14	1.42

Tabella 5.2: Distanze medie tra le poligonali della mappa ( $M_0, \dots, M_3$ ) e quelle osservate nei passaggi successivi, nel secondo test in simulazione (Sim2).

associato a  $M_2$  è dovuto principalmente alla lunghezza della poligonale che aumenta la distanza di associazione durante il confronto con le poligonali estratte nei passaggi successivi. La distanza utilizzata permette comunque di minimizzare gli effetti negativi dovuti alla differenza di lunghezza delle poligonali confrontate.

### 5.3 Risultati dei test sul robot

Le seguenti prove sono state effettuate sul robot reale, e perciò rappresentano i risultati più significativi ottenuti con l'applicazione sviluppata. La figura 5.5 mostra la mappa creata nel corridoio della palazzina 1 durante il test Pal1, con un'area di circa  $36 m^2$ , e un percorso di lunghezza  $7 m$  circa. In

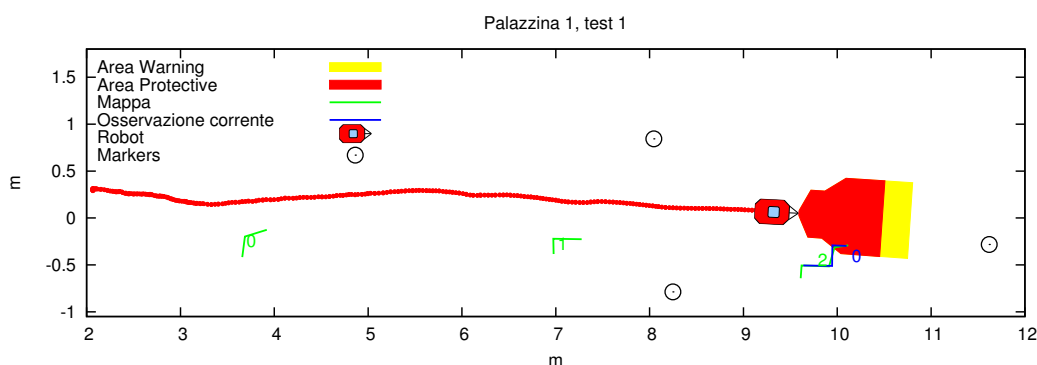


Figura 5.5: Mappa ottenuta nel corridoio della palazzina 1, con 3 ostacoli, nel test Pal1.

questo caso, e negli altri test eseguiti sul robot, gli errori commessi in termini di distanza tra le poligonali associate sono molto contenuti, come mostrato nella tabella 5.3, e permettono di riconoscere quali degli oggetti osservati sono già stati memorizzati nella mappa di riferimento. In figura 5.7 è mostrata una mappa contenente un maggior numero di poligonali, e di forma più complessa rispetto ai casi precedenti. In più la mappa ha una estensione maggiore rispetto al precedente test,  $27 m^2$ , siccome è stata ottenuta in un locale più ampio, raffigurato in 5.6.

	$M_0$	$M_1$	$M_2$
M [cm]	2.73	1.78	2.86
DS [cm]	1.11	0.99	1.16

Tabella 5.3: Risultati del il test nel corridoio della palazzina 1 (Pal1).



Figura 5.6: Ambiente in cui sono stati eseguiti i test Pal3a e Pal3b.

In questo caso alcuni oggetti statici, come muri o spigoli, sono inclusi nella mappa. Questi possono creare imprecisioni nella rappresentazione se degli oggetti quasi-statici si trovano nelle vicinanze: in alcuni casi è possibile che il sistema unisca le poligonali appartenenti ad oggetti diversi, ma questo comportamento è controllabile tramite i valori delle soglie sulle distanze utilizzate per l'associazione, la fusione e l'unione di poligonali.

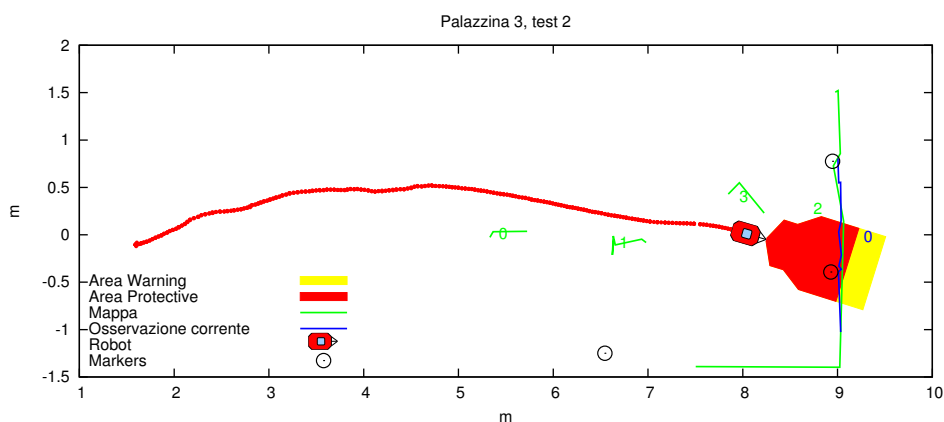


Figura 5.7: Mappa ottenuta nel corridoio della palazzina 3, con diversi ostacoli e traiettoria non rettilinea (Pal3a) .

Nella tabella 5.4 si notano valori di distanza dell'ordine dei 6 *cm* al massimo. Se si fosse utilizzata la distanza di Hausdorff definita in 3.3, alcune poligonali, come la  $M_2$  di figura 5.7, avrebbero ottenuto distanze maggiori, a causa della parziale sovrapposizione delle poligonali osservate nei passaggi successivi. Quindi la metrica impiegata risulta meno peggiorativa anche in questo caso. La figura 5.8 mostra una mappa di dimensioni simili alla pre-

	$M_0$	$M_1$	$M_2$	$M_3$
M [cm]	2.14	2.77	5.14	5.94
DS [cm]	0.70	1.28	1.55	3.18

Tabella 5.4: Risultati del test 2 nel corridoio della palazzina 3 (Pal3a).

cedente ma ottenuta con più ostacoli, con estensione di circa 31  $m^2$ , dove la traiettoria del robot è composta da più tratti curvilinei: anche in questo caso è possibile riconoscere le poligonali già osservate in precedenza con una ottima precisione.

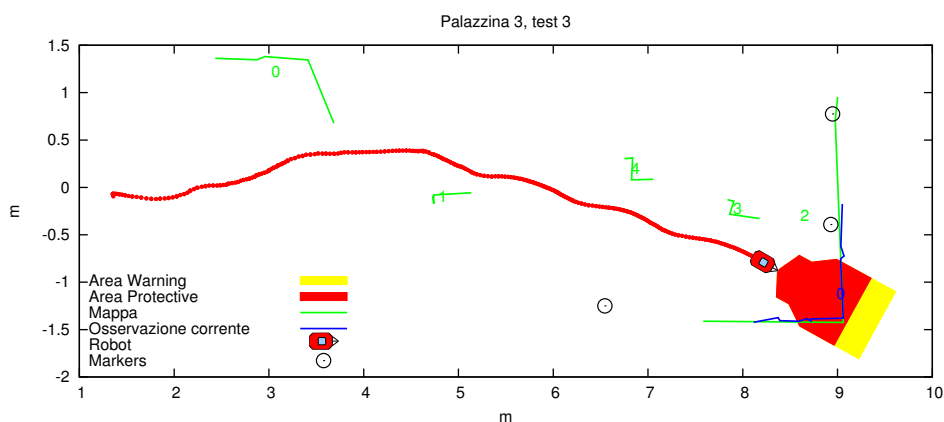


Figura 5.8: Mappa ottenuta nel corridoio della palazzina 3, secondo test, (Pal3b).

Le distanze mostrate nella tabella 5.5 sono simili a quelle ottenute negli altri casi. I risultati dei test sul robot sono paragonabili a quelli ottenuti in

	$M_0$	$M_1$	$M_2$	$M_3$	$M_4$
M [cm]	4.09	2.88	5.25	3.68	6.86
DS [cm]	3.45	1.24	1.25	1.21	1.43

Tabella 5.5: Risultati del il secondo test nel corridoio della palazzina 3 (Pal3b).

simulazione, e questo può essere dovuto all'impiego del filtro di Kalman che è in grado di migliorare la localizzazione del robot e di conseguenza ridurre l'errore di associazione tra le poligonali. La tabella 5.6 mostra le deviazioni standard del filtro associate a ciascuna delle tre componenti del vettore di stato, e in particolare i valori medi per i tre test effettuati sul robot.

In generale le poligonali che vengono osservate per un maggior numero di scansioni durante lo spostamento del robot sono quelle affette da una maggiore distanza di associazione, a causa della forma diversa (più allungata) rispetto alle poligonali presenti nella scansione corrente.

Test	Dev. standard $x$ [m]	Dev. standard $y$ [m]	Dev. standard $\theta$ [m]
Pal1	$7.079 \times 10^{-5}$	$3.029 \times 10^{-4}$	$5.774 \times 10^{-5}$
Pal3a	$7.824 \times 10^{-5}$	$3.650 \times 10^{-4}$	$1.395 \times 10^{-4}$
Pal3b	$1.126 \times 10^{-4}$	$9.681 \times 10^{-4}$	$2.333 \times 10^{-4}$

Tabella 5.6: Valori medi delle deviazioni standard associate al vettore di stato del filtro di Kalman, per i tre test effettuati sul robot.

### 5.3.1 Test in presenza di oggetti dinamici

Generalmente nei test effettuati le poligonali hanno posizioni e forme simili in tutte le prove consecutive, ma la presenza di oggetti dinamici o lo spostamento di oggetti quasi-statici tra una prova e la successiva può causare la modifica di alcune delle poligonali associate agli oggetti stessi. Per indagare questo aspetto alcuni test sono stati eseguiti in presenza di persone in movimento nello spazio di lavoro del robot durante la prova. In tali condizioni l'oggetto in movimento rispetto al sistema di riferimento della mappa è rappresentato da poligonali aventi una posizione diversa nel tempo, e tra scansioni diverse. Questo aspetto del problema può disturbare la rilevazione degli oggetti quasi-statici presenti.

A seconda delle posizioni assunte dalle persone si possono delineare alcuni situazioni:

- **occlusione:** un oggetto dinamico può occludere la visuale del sensore laser e impedire il riconoscimento di una poligonale che era stata acquisita nei transiti precedenti. In più una parziale occlusione può modificare la forma della poligonale corrente, come mostrato in figura 5.9, e determinare un aumento della distanza di associazione rilevata;

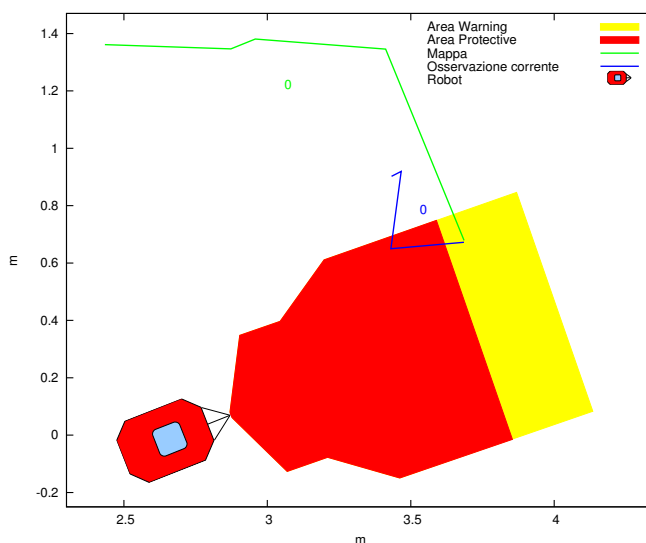


Figura 5.9: La presenza di alcuni oggetti dinamici (non raffigurati) impedisce la corretta rappresentazione della poligonale  $M_0$ .

- associazione errata: le poligonali derivate da un oggetto dinamico possono essere erroneamente associate a quelle che rappresentano oggetti statici o quasi-statici individuati nei passaggi precedenti; la figura 5.10 mostra una di queste possibili configurazioni, dove però l'algoritmo riesce ad associare correttamente la poligonale osservata con quella presente nella mappa;
- unione: nel caso in cui un oggetto dinamico si posizioni molto vicino ad oggetti già presenti nella rappresentazione corrente può accadere che le poligonali dei due oggetti vengano unite; questo può impedire il corretto riconoscimento degli oggetti nel passaggio attuale e nei successivi;
- suddivisione: più raramente può accadere che una poligonale venga divisa in più parti come conseguenza della presenza di un ostacolo dinamico in una particolare posizione, generando altri problemi di riconoscimento.

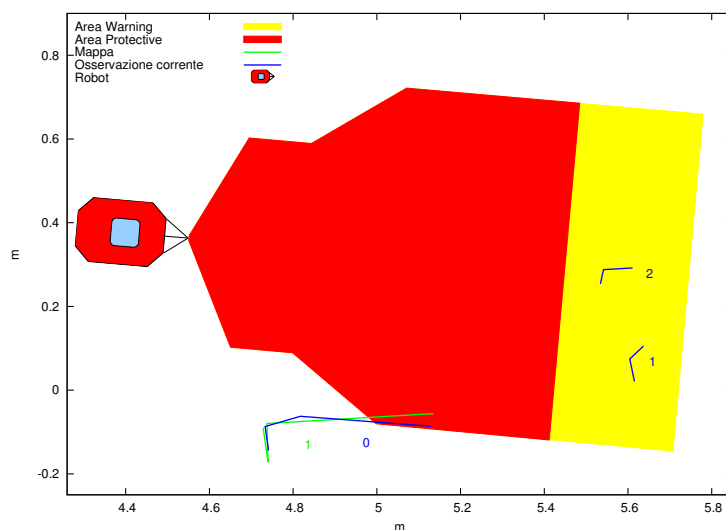


Figura 5.10: Rappresentazione di una poligonale in presenza di oggetti dinamici; si notano le gambe di una persona,  $M_1$  e  $M_2$ , in blu. La fase di associazione non viene disturbata in questo caso e la poligonale  $M_0$  blu viene correttamente associata con la  $M_1$  della mappa, in verde.

In ogni caso nei test descritti la presenza di alcuni oggetti dinamici non ha avuto effetti significativi sul riconoscimento di oggetti statici e quasi-statici. Infatti le perturbazioni sulle poligonali, misurate secondo la metrica descritta in 5.1, sono sempre state inferiori ai 10 *cm*.

Test più sistematici saranno necessari per valutare meglio gli effetti della presenza di oggetti dinamici.

## Capitolo 6

# Modello probabilistico a grafo

Il sistema realizzato in questo lavoro di tesi è indirizzato alla risoluzione di una particolare classe di problemi, che riguarda lo stoccaggio automatico di merci utilizzando LGV in ambito industriale. Le specifiche del problema e le ipotesi assunte sono state descritte in 2.1, e riguardano l'ambiente di lavoro delle navette automatiche, i requisiti di sicurezza che il sistema deve garantire, i vincoli sul movimento degli LGV e le situazioni di pericolo o intralcio al passaggio, e le possibili politiche di risoluzione di queste situazioni.

Gli LGV sono solitamente programmati per inseguire una traiettoria prestabilita che, per facilitare la progettazione, viene suddivisa in più segmenti. Questo requisito ha orientato l'attività verso l'utilizzo di rappresentazioni locali riferite ad un solo segmento di traiettoria, come già esposto in 2.5. La gestione di una mappa locale di piccole dimensioni risulta infatti più semplice, poichè su scala ridotta l'accumulo dell'errore dei dati odometrici e sensoriali non è tale da causare inconsistenze nella mappa.

Tuttavia questa scelta comporta alcuni svantaggi: in primo luogo il passaggio da un segmento di traiettoria al successivo impone la costruzione di una nuova mappa, che può essere parzialmente sovrapposta con la precedente, generando quindi dati ridondanti. Inoltre il sistema di pianificazione delle traiettorie e di navigazione degli LGV potrebbe acquisire autonomia e precisione se fosse costruita una rappresentazione globale dell'ambiente. In

questo caso le singole mappe locali, o direttamente i landmark in esse contenute, dovrebbero essere integrate in una unica mappa globale, operazione che comporta lo sviluppo di un algoritmo in grado di allineare le diverse componenti. Tale soluzione modifica però profondamente l'architettura del sistema di LGV considerato in questa tesi.

A completamento del lavoro si è ritenuto opportuno indagare questa opzione più complessa. Infatti la rappresentazione locale potrebbe non essere adatta ad un contesto meno strutturato di quello analizzato, nel quale le traiettorie degli LGV potrebbero non essere segmentate, o persino non definite a priori, e quindi sarebbe più complicato scegliere la giusta dimensione per le mappe locali e la frequenza del loro aggiornamento. In più potrebbe non essere disponibile un sistema di localizzazione del robot, e in questo caso il sistema dovrebbe provvedere alla localizzazione e alla costruzione della mappa simultaneamente.

La soluzione studiata in questa parte del lavoro prevede la costruzione della mappa globale dell'ambiente composta da features poligonali, supportata da una struttura a grafo per la rappresentazione e da un modello probabilistico per considerare l'incertezza associata ai dati sensoriali. Il modello SP, introdotto in 6.2, viene esteso per integrare le poligonali tramite un framework probabilistico adeguato alla risoluzione di problemi di SLAM, chiamato SP-map e descritto in 6.3. Successivamente, in 6.3.1, viene descritta l'integrazione delle SPmap nel modello a grafo utilizzato per la creazione e la risoluzione della mappa.

## 6.1 SLAM

Il problema dello SLAM è già stato introdotto nel capitolo 1, e consiste nella creazione della mappa dell'ambiente durante il movimento del robot, e, simultaneamente, nella localizzazione del robot. Un metodo intuitivo e efficiente per rappresentare gli elementi costitutivi di un problema di SLAM consiste nell'uso di una struttura a grafo come supporto per la mappa, insieme ad un

modello probabilistico per la stima dei dati provenienti dai sensori. Il grafo costruito può successivamente essere “risolto”, procedimento tramite il quale si può migliorare la precisione e fedeltà della ricostruzione dell’ambiente osservato.

### 6.1.1 Formulazione probabilistica dello SLAM

L’incertezza associata ai dati sensoriali, sia provenienti da laser scanner che da altri tipi di sensori, viene considerata nel modello sensoriale tipico dello SLAM tramite l’uso della teoria della probabilità. Il movimento del robot nell’ambiente viene descritto come una traiettoria rappresentata da una sequenza di variabili aleatorie  $\mathbf{x}_{1:T} = \{\mathbf{x}_1, \dots, \mathbf{x}_T\}$ , da una serie di valori di odometria del robot, indicata con  $\mathbf{u}_{1:T} = \{\mathbf{u}_1, \dots, \mathbf{u}_T\}$ , e dalle osservazioni sensoriali avvenute, indicate con  $\mathbf{z}_{1:T} = \{\mathbf{z}_1, \dots, \mathbf{z}_T\}$ . La risoluzione del problema di SLAM implica la stima della distribuzione di probabilità associata ad  $\mathbf{x}$  e alla mappa  $\mathbf{m}$  date le osservazioni  $\mathbf{z}$ , le misurazioni di odometria  $\mathbf{u}$  e una posizione iniziale  $\mathbf{x}_0$

$$p(\mathbf{x}_{1:T}, \mathbf{m} \mid \mathbf{z}_{1:T}, \mathbf{u}_{1:T}, \mathbf{x}_0) \quad (6.1)$$

Per stimare questa probabilità è conveniente descrivere il problema dello SLAM attraverso una struttura composta da un grafo orientato, dove ogni nodo rappresenta una variabile aleatoria, e ogni vertice rappresenta una relazione di dipendenza tra i nodi connessi. In figura 6.1 è mostrata la rappresentazione grafica di questa struttura, chiamata *Rete Bayesiana Dinamica* (DBN), la quale descrive il processo stocastico di stima della traiettoria  $\mathbf{x}_t$ . Gli archi tra i nodi  $\mathbf{x}_t$  e  $\mathbf{x}_{t-1}$  rappresentano il modello di transizione dello stato (la posa del robot stimata) espresso come  $p(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \mathbf{u}_t)$ , cioè la probabilità che il robot si trovi in  $\mathbf{x}_t$  al tempo  $t$ , dato che si trovava in  $\mathbf{x}_{t-1}$  al tempo  $t - 1$ , e ha rilevato un valore di odometria pari a  $\mathbf{u}_t$ . Gli archi tra i nodi  $\mathbf{x}_t$  e  $\mathbf{z}_t$  rappresentano invece il modello delle osservazioni,  $p(\mathbf{z}_t \mid \mathbf{x}_t, \mathbf{m}_t)$ , cioè la probabilità di effettuare le osservazioni  $\mathbf{z}_t$  quando il robot si trova in  $\mathbf{x}_t$  e dispone della mappa  $\mathbf{m}_t$ .

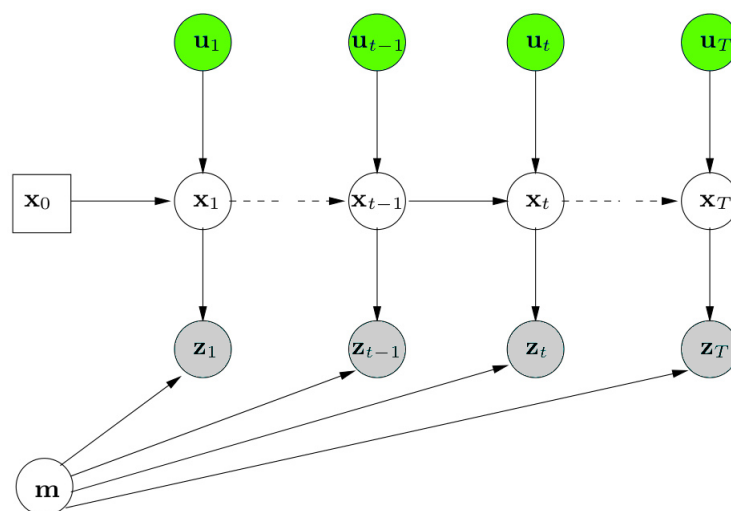


Figura 6.1: Rappresentazione grafica di un problema di SLAM: la rete bayesiana mostra i nodi associati alle variabili aleatorie  $\mathbf{x}_t$ ,  $\mathbf{z}_t$ ,  $\mathbf{u}_t$ ,  $\mathbf{m}_t$ , e le relazioni di dipendenza tra di loro.

### 6.1.2 Formulazione a grafo dello SLAM

Una rappresentazione alternativa per il problema dello SLAM è quella basata su grafi, già descritta in 1.1 e chiamata *Graph-Based SLAM* [20], in cui sono modellate esplicitamente le relazioni spaziali tra le variabili coinvolte. Le pose del robot sono rappresentate come nodi nel grafo a cui è associata la loro posizione nello spazio, così come le osservazioni sensoriali, che vengono collegate tramite gli archi ai nodi posa da cui sono state effettuate. Gli archi rappresentano quindi dei vincoli spaziali, che consistono in una distribuzione di probabilità sulla trasformazione relativa tra i due nodi collegati. In figura 6.2 è rappresentato schematicamente un grafo di questo tipo. In questi termini il problema da risolvere viene suddiviso in due sottoproblemi: la costruzione del grafo partendo dai dati sensoriali acquisiti, con lo scopo di inserire nel grafo nodi e archi che sono la base per la seconda fase, cioè la risoluzione (o ottimizzazione) del grafo per determinare la configurazione di nodi che meglio si adatta ai vincoli presenti, e quindi risulta essere quella più probabile. La fase di costruzione dipende dal tipo di sensore e di rap-

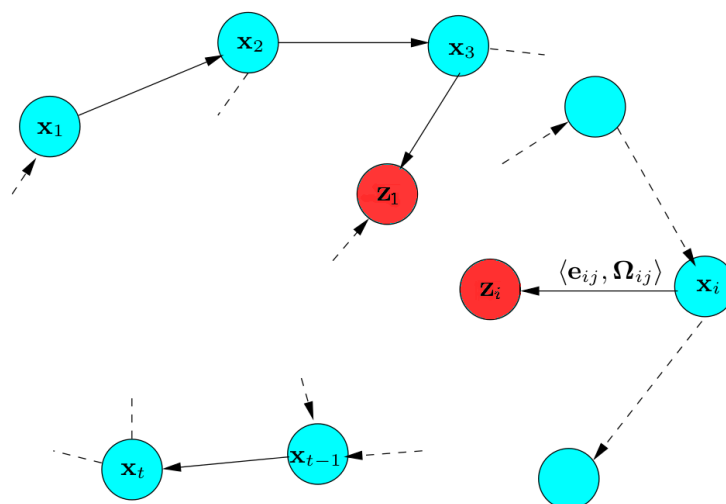


Figura 6.2: Rappresentazioni schematica di un grafo di pose e landmark. I nodi etichettati con  $\mathbf{x}_i$  denotano le pose del robot, mentre quelli etichettati con  $\mathbf{z}_i$  denotano le osservazioni sensoriali.

presentazione scelti per la specifica applicazione, mentre l'ottimizzazione è indipendente dal contenuto e dalla struttura del grafo.

Uno dei maggiori problemi da considerare durante la costruzione del grafo è rappresentato dall'associazione dei dati osservati con quelli già presenti nel grafo stesso, operazione chiamata data association. Questa operazione è necessaria anche nella costruzione di mappe locali, ma risulta ancora più complessa nel caso della mappe globali. Quando il robot visita una zona dell'ambiente già visitata in precedenza è necessario individuare e riconoscere i nodi corrispondenti agli elementi correntemente osservati e associarli ad essi. Questa operazione porta a connettere lo stesso nodo (landmark) a più nodi che rappresentano una posa del robot. La relazione implicita tra due pose consente di evidenziare inconsistenze nel valore corrente della pose stesse. La configurazione del grafo ritenuta più verosimile è quella che minimizza l'errore inerente ai vincoli introdotti dalle osservazioni. Tale errore è rappresentato da una funzione di verosimiglianza (o *likelihood*) che deve essere stimata mediante la procedura di risoluzione del grafo. È quindi indispensabile valutare la media e la matrice informazione della distribuzione di

probabilità Gaussiana che meglio approssima la traiettoria del robot a posteriori. Solitamente si cerca di effettuare questo processo di stima attraverso la risoluzione di un problema di ottimizzazione vincolata: data una serie di nodi  $\mathbf{x} = [\mathbf{x}_1, \dots, \mathbf{x}_T]^T$ , si definisce la media  $\mathbf{z}_{ij}$  e la matrice informazione  $\mathbf{\Omega}_{ij}$  del vincolo virtuale tra nodi  $i$  e  $j$ . Questo vincolo rappresenta la trasformazione che massimizza la probabilità dell'osservazione  $j$  effettuata dal nodo  $i$ . Inoltre, data la configurazione di due nodi  $\mathbf{x}_i$  e  $\mathbf{x}_j$ , la predizione del vincolo  $(i, j)$  è definita come  $\hat{\mathbf{z}}_{ij}(\mathbf{x}_i, \mathbf{x}_j)$ , e rappresenta solitamente la trasformazione relativa tra i sistemi di riferimento associati ai due nodi. Per convenienza, invece di operare sulla funzione di verosimiglianza (che è Gaussiana) viene introdotta la *loglikelihood negativa* dell'osservazione  $\mathbf{z}_{ij}$ , definita da

$$l_{ij} \propto [\mathbf{z}_{ij} - \hat{\mathbf{z}}_{ij}(\mathbf{x}_i, \mathbf{x}_j)]^T \mathbf{\Omega}_{ij} [\mathbf{z}_{ij} - \hat{\mathbf{z}}_{ij}(\mathbf{x}_i, \mathbf{x}_j)] \quad (6.2)$$

dove  $[\mathbf{z}_{ij} - \hat{\mathbf{z}}_{ij}(\mathbf{x}_i, \mathbf{x}_j)] = \mathbf{e}_{ij}(x_i, x_j)$  rappresenta la stima dell'errore di misurazione tra l'osservazione predetta  $\hat{\mathbf{z}}_{ij}$  e la vera osservazione  $\mathbf{z}_{ij}$  acquisita dal sensore. La figura 6.3 mostra i nodi e i vincoli che rientrano nella definizione di un vertice del grafo e della sua loglikelihood. Definito  $\mathcal{C}$  l'insieme di

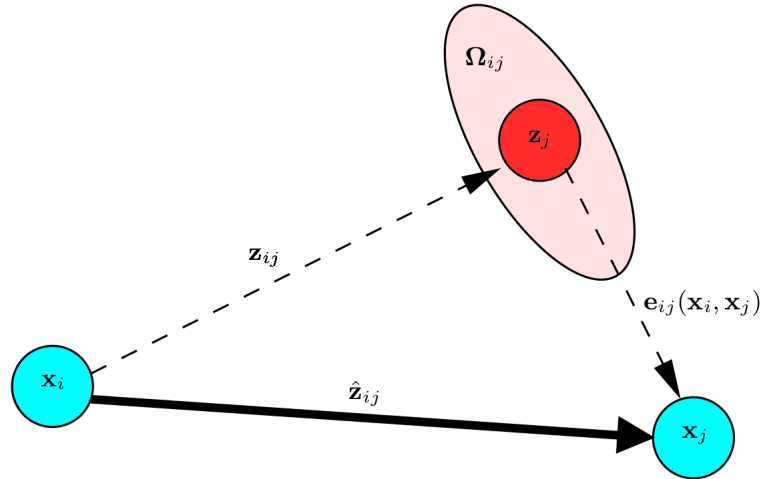


Figura 6.3: Vincolo che connette i nodi  $\mathbf{x}_i$  e  $\mathbf{x}_j$ , generato dall'osservazione reale  $\mathbf{z}_{ij}$ . Tramite la trasformazione relativa tra i nodi si ottiene l'osservazione predetta  $\hat{\mathbf{z}}_{ij}$ , che rappresenta  $\mathbf{x}_j$  rispetto alla terna associata ad  $\mathbf{x}_i$ .

coppie di indici dei nodi per i quali esiste un vertice dato da una osserva-

zione, l'obiettivo della fase di risoluzione del grafo consiste nel trovare la configurazione di nodi che minimizza la loglikelihood negativa definita da

$$\mathbf{F}(\mathbf{x}) = \sum_{\langle i,j \rangle \in \mathcal{C}} \mathbf{e}_{ij}^T \boldsymbol{\Omega}_{ij} \mathbf{e}_{ij} \quad (6.3)$$

cioè trovare la configurazione  $\mathbf{x}^*$  tale che

$$\mathbf{x}^* = \underset{\mathbf{x}}{\operatorname{argmin}} \mathbf{F}(\mathbf{x}) \quad (6.4)$$

La soluzione di questo problema di ottimizzazione può essere ottenuta tramite diverse tecniche e algoritmi; in questo lavoro di tesi è stata utilizzata la tecnica di ottimizzazione di *Gauss-Seidel*, descritta in [23], per la risoluzione della rete di vincoli.

## 6.2 Introduzione all'SPmodel

La rappresentazione dell'incertezza associata alle osservazioni sensoriali, e di conseguenza alle entità geometriche o features descritte da questi dati, deve essere adeguatamente modellata per garantire un vantaggio a livello applicativo. A questo scopo è stato studiato un modello probabilistico chiamato *Symmetries and Perturbation Model* (SPModel) [26], che integra le informazioni geometriche provenienti da uno o più sensore e, considerando la loro incertezza, consente di stimare la locazione delle features estratte.

L'SPmodel combina l'uso della teoria della probabilità per rappresentare l'imprecisione nella posizione di un elemento geometrico, e la teoria delle simmetrie per rappresentare la parzialità della visuale tipica di un sensore di prossimità. Il concetto di simmetria, cioè una trasformazione che preserva l'oggetto a cui è applicata, è raffigurato in 6.4: un punto individuato sul bordo di un oggetto osservato possiede due gradi di libertà. In particolare questi sono la posizione sul bordo, e l'orientazione attorno al bordo stesso. Inoltre esistono due soluzioni alternative, corrispondenti ai due bordi dell'oggetto. Quindi si può affermare che il bordo rappresenta un feature che possiede simmetria trasazionale su di essa, simmetria rotazionale attorno

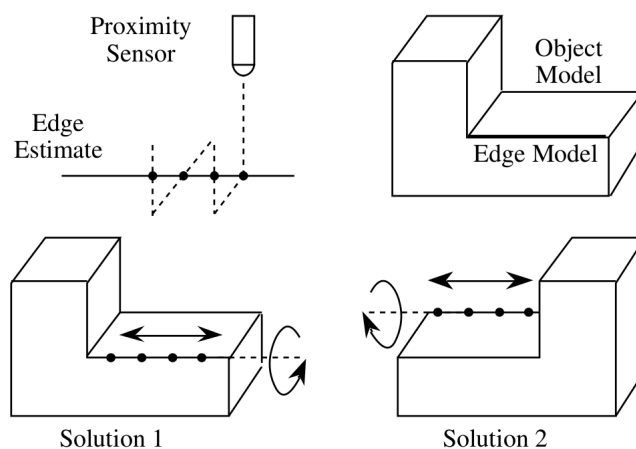


Figura 6.4: Simmetrie possibili per un bordo di un oggetto osservato da un sensore di prossimità, e quindi rappresentato tramite una vista parziale (immagine tratta da [26]).

ad essa, e simmetria ciclica attorno ad ogni asse perpendicolare ad essa. Le simmetrie quindi rappresentano tutte le possibili trasformazioni tra le terne che descrivono al stessa locazione di un oggetto sul piano di scansione. Queste trasformazioni sono indicate con  $t_{AB}$  in figura 6.5, e sono composte dalla posa relativa tra i due sistemi di riferimento  $A$  e  $B$ .

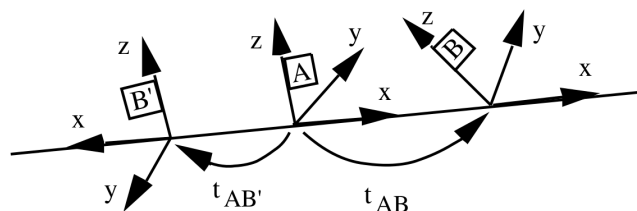


Figura 6.5: Simmetrie possibili per una retta infinita: tutte le trasformazioni  $t$  danno luogo ad un punto che risiede ancora sulla retta (immagine tratta da [26]).

### 6.2.1 Corrispondenza tra features

Il modello SPmodel associa ad ogni feature un sistema di riferimento,  $E$ , i cui assi devono essere allineati con la feature, come mostrato nella illustrazione 6.5 nel caso di una linea, dove l'asse  $x$  è allineato con la retta stessa. La

feature è rappresentata tramite le sue coordinate globali rispetto ad una terna di riferimento  $W$  dal vettore di posizione  $\mathbf{x}_{WE} = [x, y, \theta]$ , e da una *matrice di binding* (o di selezione), cioè una matrice le cui righe selezionano quali coordinate del vettore di posizione sono soggette a simmetria. La tabella 6.1 riporta alcune matrici di selezione per le features più semplici. L'utilizzo

Feature	Matrice di selezione
Punto	$\mathbf{B} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$
Retta infinita	$\mathbf{B} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$
Posa del robot	$\mathbf{B} = \mathbf{I}_3$

Tabella 6.1: Matrici di selezione per le features più comuni.

di queste matrici consente di esprimere la *corrispondenza* di due features in modo del tutto generale, indipendente dal tipo di feature in esame. Questa possibilità è uno dei maggiori contributi introdotti dal SPmodel, che facilita la fase di associazione delle features e di riconoscimento degli oggetti, evitando i problemi derivati da differenti rappresentazioni e dalle singularità che alcune di queste rappresentazioni possono avere.

Date due terne  $A$  e  $B$  associate ad altrettante features, la corrispondenza è esprimibile dall'equazione 6.5

$$\mathbf{B}_{AB} \mathbf{x}_{AB} = 0 \quad (6.5)$$

dove  $\mathbf{B}_{AB}$  contiene solo le righe comuni alle matrici di selezione associate alle features ( $\mathbf{B}_A$  e  $\mathbf{B}_B$ ) e  $\mathbf{x}_{AB}$  è la trasformazione relativa.

La figura 6.6 mostra un esempio di calcolo della corrispondenza tra una linea infinita  $E$  e un punto  $P$ . Denominata  $\mathbf{x}_{EP} = [x_{EP}, y_{EP}, \theta_{EP}]^T$  la posizione del punto  $P$  relativa alla terna  $E$  della retta, utilizzando le corrette matrici di selezione elencate nella tabella 6.1 si può costruire la matrice di selezione della corrispondenza tra  $E$  e  $P$ ,  $\mathbf{B}_{EP} = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}$ , che permette di determinare

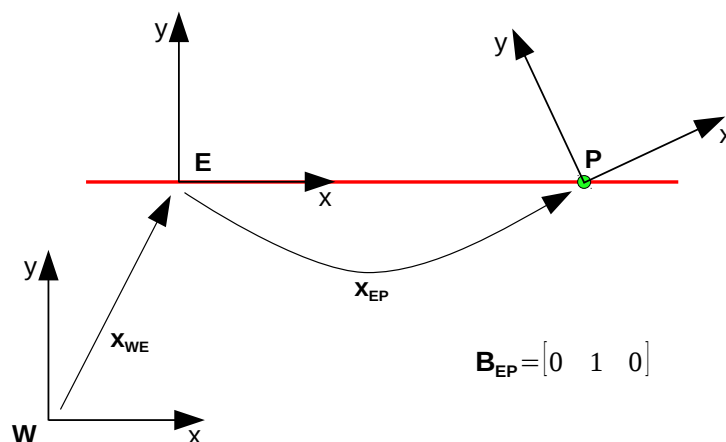


Figura 6.6: Matrici di selezione  $\mathbf{B}_{EP}$  della corrispondenza tra le features  $E$  (linea infinita) e  $P$  (punto).

l'equazione della corrispondenza

$$\mathbf{B}_{EP} \mathbf{x}_{EP} = 0 \quad ; \quad y_{EP} = 0 \quad (6.6)$$

L'equazione 6.6 esprime l'appartenenza del punto  $P$  alla retta solo se la sua posizione sull'asse  $y$  rispetto alla terna  $E$  è uguale a zero.

## 6.2.2 Rappresentazione dell'incertezza

Nella classica formulazione dei modelli probabilistici l'incertezza legata all'uso di un sensore è rappresentata da rumore bianco con densità di probabilità Gaussiana, caratterizzata dalla sua media e dalla sua matrice di covarianza. Nel SPmodel viene impiegata una rappresentanza locale di incertezza: la posizione stimata di una feature  $F$  rispetto ad una terna di riferimento  $W$ , è denotata da  $\hat{\mathbf{x}}_{WF}$ , la quale rappresenta la migliore approssimazione attualmente disponibile della reale posizione. L'errore di stima è rappresentato localmente da un *vettore di perturbazione*, applicato rigidamente alla feature. La reale posizione dell'elemento  $F$  è data dalla composizione

$$\mathbf{x}_{WF} = \hat{\mathbf{x}}_{WF} \oplus \mathbf{d}_F \quad (6.7)$$

dove il vettore  $\mathbf{d}_F = [d_x, d_y, d_\theta]^T$  viene chiamato vettore differenziale di posizione, e rappresenta l'errore sulla posizione della feature: le sue componenti che corrispondono ai gradi di libertà della feature sono nulle, siccome non rappresentano un effettivo errore. Il vettore di perturbazione  $\mathbf{p}_F$  viene ottenuto rimuovendo da  $\mathbf{d}_F$  le componenti nulle, tramite l'uso della matrice di selezione (eq. 6.8).

$$\mathbf{p}_F = \mathbf{B}_F \mathbf{d}_F \quad ; \quad \mathbf{d}_F = \mathbf{B}_F^T \mathbf{p}_F \quad (6.8)$$

Quindi l'SPmodel rappresenta la posizione di una feature tramite la quadrupla

$$\mathbf{L}_{WF} = (\hat{\mathbf{x}}_{WF}, \hat{\mathbf{p}}_F, \mathbf{C}_F, \mathbf{B}_F) \quad (6.9)$$

dove  $\hat{\mathbf{x}}_{WF}$  è la stima della posizione usata come terna di riferimento per il valore predetto  $\hat{\mathbf{p}}_F$  del vettore di perturbazione, e  $\mathbf{C}_F$  è la covarianza di questa variabile aleatoria. Quando  $\hat{\mathbf{p}}_F = 0$  la stima si definisce *centrata*.

Il principale vantaggio di questa rappresentazione è la sua generalità, infatti può essere utilizzata con qualunque tipo di feature estratta da una qualunque osservazione sensoriale. Inoltre non sussistono problemi di singolarità, e di conseguenza le matrici di covarianza sono sempre ben definite, il modello non è sovra-parametrizzato e quindi non esistono parametri dal significato o dal valore non determinato, e l'incertezza rappresentata dal vettore di perturbazione non dipende dalla terna di base del mondo ( $W$ ).

### 6.3 Un framework probabilistico per il problema di SLAM: SPmap

L'utilizzo del SPmodel permette di rappresentare qualunque tipo di feature in modo generale, e, se combinato con una strategia di integrazione delle osservazioni sensoriali, consente anche di stimare la locazione di una feature o di un oggetto complesso partendo da un insieme di osservazioni affette da incertezza, che possono interessare più viste diverse dell'oggetto.

Le proprietà indispensabili di un sistema che lavora nel contesto dello SLAM riguardano la precisione nella ricostruzione della traiettoria del robot e della mappa dell'ambiente. Queste caratteristiche si possono ottenere grazie all'utilizzo di una adeguata rappresentazione degli oggetti osservati e grazie ad una procedura di localizzazione affidabile; per questi motivi è conveniente integrare le features rappresentate tramite l'SPmodel in un framework probabilistico chiamato *SPmap* [27]. Questo framework descrive un sistema completo e rigoroso per la soluzione di un problema di SLAM, le sue caratteristiche principali sono:

- rappresentazione consistente e integrazione delle osservazioni sensoriali, anche se acquisite da diversi tipi di sensori;
- assenza di problemi di singolarità nella rappresentazione utilizzata (SPmodel);
- possibilità di integrazione in una struttura a grafo per la risoluzione della rete di vincoli.

Nell'SPmap la composizione della mappa include la posizione attuale del robot descritta secondo l'equazione 6.9 del SPmodel,  $\mathbf{L}_{WR}$ , e la posizione di ognuna delle features,  $\mathbf{L}_{WF_i}$

$$\mathbf{L}_{WR} = (\hat{\mathbf{x}}_{WR}, \hat{\mathbf{p}}_R, \mathbf{C}_R, \mathbf{B}_R) \quad (6.10)$$

$$\mathbf{L}_{WF_i} = (\hat{\mathbf{x}}_{WF_i}, \hat{\mathbf{p}}_{F_i}, \mathbf{C}_{F_i}, \mathbf{B}_{F_i}) \quad i = 1, \dots, N_F \quad (6.11)$$

La figura 6.7 mostra un esempio di configurazione di robot e features rappresentate dall'SPmap. Sinteticamente la mappa si può descrivere tramite la quadrupla

$$SPmap = (\hat{\mathbf{x}}^W, \mathbf{p}^W, \mathbf{C}^W, \mathbf{B}^W) \quad (6.12)$$

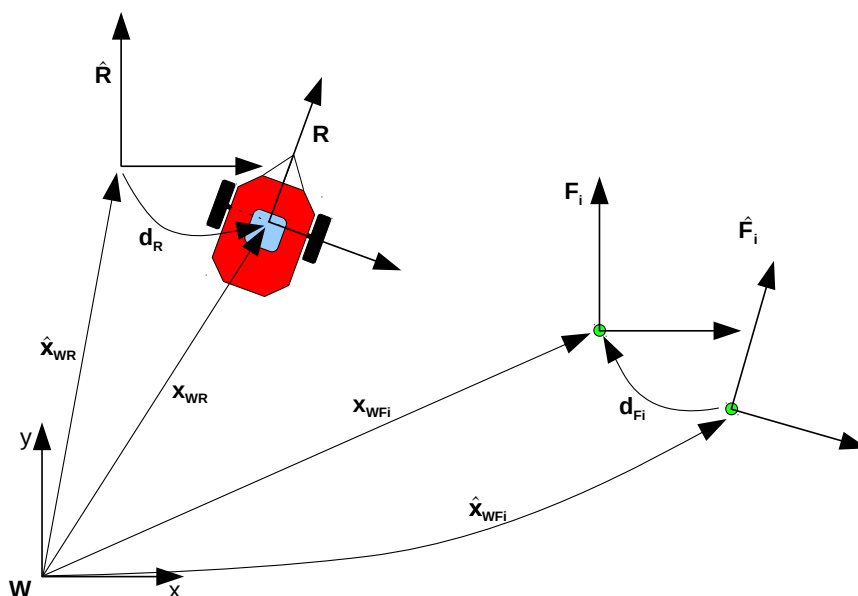


Figura 6.7: Rappresentazione grafica del robot e di una feature secondo il modello SPmap.

dove  $\hat{\mathbf{x}}^W$  è il vettore delle posizioni stimate della mappa (robot e features), mentre  $\mathbf{p}^W$  rappresenta i vettori di perturbazione.

$$\hat{\mathbf{x}}^W = \begin{bmatrix} \hat{\mathbf{x}}_{WR} \\ \hat{\mathbf{x}}_{WF_1} \\ \vdots \\ \hat{\mathbf{x}}_{WF_{N_F}} \end{bmatrix} ; \quad \mathbf{p}^W = \begin{bmatrix} \mathbf{d}_R \\ \mathbf{p}_{F_1} \\ \vdots \\ \mathbf{p}_{F_{N_F}} \end{bmatrix} \quad (6.13)$$

La reale locazione del robot e delle features viene ottenuta tramite

$$\mathbf{x}^W = \hat{\mathbf{x}}^W \oplus (\mathbf{B}^W)^T \mathbf{p}^W \quad (6.14)$$

dove la composizione deve esser applicata per ogni riga del vettore  $\mathbf{x}^W$ , e  $\mathbf{B}^W$  è la matrice di selezione diagonale dell'intera mappa, composta dalle matrici di selezione del robot e delle singole features

$$\mathbf{B}^W = \text{diag}(\mathbf{B}_R, \mathbf{B}_{F_1}, \dots, \mathbf{B}_{F_{N_F}}). \quad (6.15)$$

La matrice di covarianza della mappa  $\mathbf{C}^W$  è composta dalle matrici di covarianza del processo di stima delle posizioni di robot e features, e da tutte le

matrici di covarianza incrociata tra robot e features e tra le features

$$\mathbf{C}^W = \begin{bmatrix} \mathbf{C}_R & \mathbf{C}_{RF_1} & \cdots & \mathbf{C}_{RF_{N_F}} \\ \mathbf{C}_{RF_1}^T & \mathbf{C}_{F_1} & \cdots & \mathbf{C}_{F_1 F_{N_F}} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{C}_{RF_{N_F}}^T & \mathbf{C}_{F_1 F_{N_F}}^T & \cdots & \mathbf{C}_{F_{N_F}} \end{bmatrix} \quad (6.16)$$

### 6.3.1 Integrazione nel modello a grafo

La mappa descritta dal modello SPmap può essere integrata in una struttura a grafo che permette di ricondurre il problema dell'ottimizzazione della mappa ad un tipico problema di ottimizzazione vincolata, per il quale esistono molti metodi di risoluzione. L'approccio introdotto in questa sezione prende il nome di *maximum likelihood* (ML), e deriva da una strategia di stima della mappa chiamata *maximum a posteriori* (MAP): date le pose del robot  $x_{0:T}$ , le features della mappa  $f_{1:M}$ , i valori dell'odometria  $u_{1:T}$  e le osservazioni sensoriali  $z_{1:T}$ , la likelihood della mappa può essere riscritta in questo caso come  $p(x_{0:T}, f_{1:M} | u_{1:T}, z_{1:T})$ . Come già descritto in 6.1.1 e 6.1.2 il processo di stima della mappa mira a trovare la configurazione dei nodi del grafo che massimizza la likelihood di equazione 6.1, cioè, in questo caso

$$[x_{0:T}, f_{1:M}]^* = \operatorname{argmax}_{x_{0:T}, l_{1:M}} p(x_{0:T}, l_{1:M} | u_{1:T}, z_{1:T}) \quad (6.17)$$

La distribuzione di probabilità 6.17 non è semplice da stimare, ma può essere scomposta secondo il teorema di Bayes e l'ipotesi di Markov, per dar luogo ad una formulazione più facilmente trattabile

$$p(x_{0:T}, f_{1:M} | u_{1:T}, z_{1:T}) = \eta p(x_0) \prod_{t=0}^T p(z_t | x_t, f_{\mathcal{O}_t}) p(x_t | x_{t-1}, u_t) \quad (6.18)$$

dove  $\mathcal{O}_t$  rappresenta l'insieme di features osservate al tempo  $t$ , ed associate alle features della mappa secondo la procedura di associazione descritta nel capitolo 3, in 3.2.

Le densità di probabilità  $p(x_t | x_{t-1}, u_t)$  e  $p(z_t | x_t, f_{\mathcal{O}_t})$  rappresentano rispettivamente il moto del robot e le osservazioni effettuate, e sono descritte dal

modello del moto  $x_t$  e il modello sensoriale  $z_t$  utilizzati:

$$x_t = g(x_{t-1}, u_t) + w_t \quad (6.19)$$

$$z_t = h(x_t, f_{\mathcal{O}_t}) + v_t \quad (6.20)$$

dove  $g(\cdot)$  ed  $h(\cdot)$  sono funzioni deterministiche (spesso non lineari) e i termini  $w_t$  e  $v_t$  rappresentano rumore additivo gaussiano bianco, le cui covarianze sono indicate rispettivamente con  $R_t$  e  $Q_t$ . Dal momento che una densità di probabilità Gaussiana è espressa nella classica forma esponenziale, è conveniente utilizzare la loglikelihood negativa, al fine di rimuovere l'esponenziale. Quindi il problema descritto dall'equazione 6.17 viene riformulato come la stima della configurazione dei nodi che minimizza la loglikelihood negativa di espressione

$$\begin{aligned} -\log p(x_{0:t}, f_{1:M} | z_{1:t}, u_{1:t}) &= \text{const.} + \frac{1}{2} x_0^T R_0^{-1} x_0 \\ &+ \frac{1}{2} \sum_t (x_t - g(x_{t-1}, u_t))^T R_t^{-1} (x_t - g(x_{t-1}, u_t)) \\ &+ \frac{1}{2} \sum_t \sum_i (z_t^i - h(x_t, f_t^i))^T Q_t^{-1} (z_t^i - h(x_t, f_t^i)) \end{aligned} \quad (6.21)$$

Il termine  $x_t - g(x_{t-1}, u_t)$  esprime un vincolo tra  $x_{t-1}$  e  $x_t$  rappresentato dal valore dell'odometria  $u_t$ . Similmente il termine  $z_t - h(x_t, f_t^i)$  esprime un vincolo tra l'osservazione  $f_t^i$ , e la posa del robot  $x_t$  dalla quale è stata acquisita. Queste relazioni tra le variabili vengono rappresentate come degli archi nel grafo, e modellano i vincoli tra i nodi che rappresentano le variabili del sistema.

Nel modello SPmap il vincolo tra due nodi è descritto dalla decomposizione dei sistemi di riferimento associati ai nodi stessi. Questo vale per qualunque tipo di nodo, posa o feature, ed anche per qualunque tipo di feature, grazie alla proprietà di generalità offerta dalle features SP. In figura 6.8 è mostrato un esempio di SPmap supportata da una struttura a grafo: si può notare la presenza di due feature diverse, un punto e un segmento, che possono essere inserite nella mappa e nella rete di vincoli utilizzando le corrette matrici di selezione, associate alle feature stesse secondo la tabella 6.1.

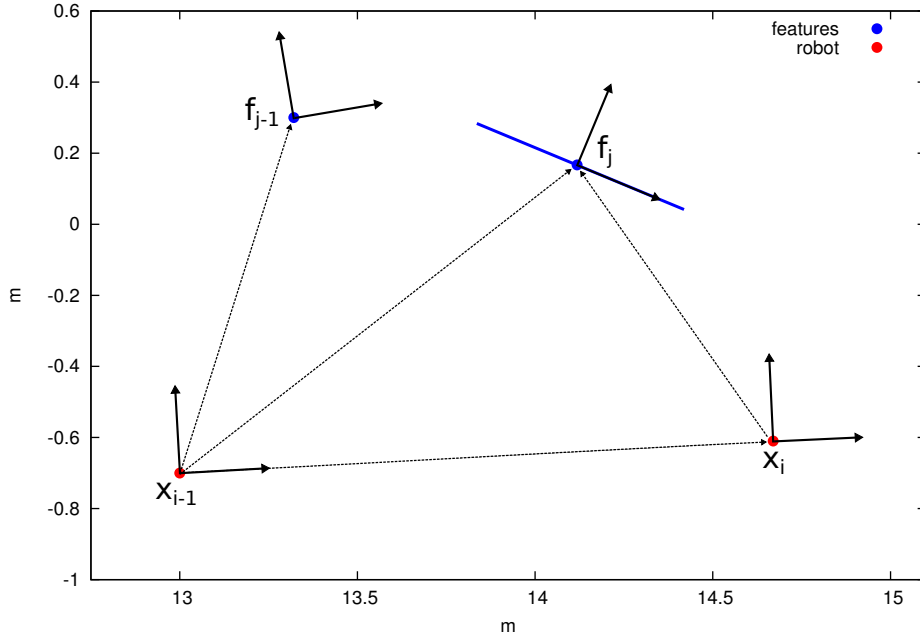


Figura 6.8: Esempio di modello a grafo della SPmap in cui si possono notare due pose del robot e due features, una puntiforme, e una rappresentata da un segmento.

Dati un nodo sorgente  $\mathbf{s}$  e un nodo destinazione  $\mathbf{d}$  il vincolo è dato da

$$d(\mathbf{s}, \mathbf{d}) = \mathbf{s} \ominus \mathbf{d} \quad (6.22)$$

la cui espressione completa è mostrata nell'appendice B. Utilizzando le equazioni del modello SPmap le posizioni stimate dei nodi  $\mathbf{s}$  e  $\mathbf{d}$  sono

$$\mathbf{s} = \hat{\mathbf{s}} \oplus \mathbf{B}_s^T \mathbf{p}_s \quad ; \quad \mathbf{d} = \hat{\mathbf{d}} \oplus \mathbf{B}_d^T \mathbf{p}_d. \quad (6.23)$$

Quindi, chiamati  $\mathbf{d}_s = \mathbf{B}_s^T \mathbf{p}_s$  e  $\mathbf{d}_d = \mathbf{B}_d^T \mathbf{p}_d$ , sostituendo  $\mathbf{s}$  e  $\mathbf{d}$  in 6.22 la funzione di vincolo generale risulta essere

$$f(\mathbf{s}, \mathbf{d}, \mathbf{d}_s, \mathbf{d}_d) = (\mathbf{s} \oplus \mathbf{d}_s) \ominus (\mathbf{d} \oplus \mathbf{d}_d) = d(c(\mathbf{s}, \mathbf{d}_s), c(\mathbf{d}, \mathbf{d}_d)) \quad (6.24)$$

dove  $c(\cdot)$  denota la composizione di due sistemi di riferimento.

### 6.3.2 Risoluzione del grafo

I metodi di risoluzione solitamente utilizzati nel Graph-SLAM sono basati sulla linearizzazione della funzione di vincolo e sulla soluzione del sistema

lineare di equazioni risultante per tutto il grafo. L'espressione della funzione di vincolo di equazione 6.24 contiene termini non lineari a causa delle rotazione presente nella composizione e nella decomposizione, e quindi viene linearizzata, tramite espansione di Taylor attorno al punto  $(\hat{\mathbf{s}}, \hat{\mathbf{d}}, 0, 0)$ :

$$f(\mathbf{s}, \mathbf{d}, \mathbf{d}_s, \mathbf{d}_d) \approx f(\hat{\mathbf{s}}, \hat{\mathbf{d}}, 0, 0) + \mathbf{F}_{sd}^s(\mathbf{s} - \hat{\mathbf{s}}) + \mathbf{F}_{sd}^d(\mathbf{d} - \hat{\mathbf{d}}) \quad (6.25)$$

dove gli jacobiani  $\mathbf{F}_{sd}^s$  e  $\mathbf{F}_{sd}^d$  sono derivati dagli jacobiani delle funzioni di composizione e decomposizione, come mostrato nell'appendice B, da B.8 a B.11. Dopo la linearizzazione la loglikelihood negativa risulta essere una funzione quadratica rispetto alle variabili di stato rappresentate dai nodi del grafo  $\mathbf{s}_i$ ,  $i = 0, \dots, n$  (sorgente di un vertice) e  $\mathbf{d}_j$ ,  $j = 0, \dots, m$  (destinazione di un vertice). Dato l'insieme  $\mathcal{C}$  di tutti i vincoli, i valori medi dei vincoli  $\boldsymbol{\mu}_{ij}$  e le matrici informazione  $\boldsymbol{\Omega}_{ij}$  di ogni vincolo tra i nodi  $i$  e  $j$ , i vettori di posizione differenziale  $\mathbf{d}_{s_i}$  per i nodi sorgente e  $\mathbf{d}_{d_j}$  per i nodi destinazione, e un punto di linearizzazione con valori delle variabili di stato  $\hat{\mathbf{s}}_i$  e  $\hat{\mathbf{d}}_j$ , la funzione loglikelihood negativa è definita dalla seguente espressione

$$\begin{aligned} \chi^2 &= \sum_{\langle i,j \rangle \in \mathcal{C}} (f(\mathbf{s}_i, \mathbf{d}_j, \mathbf{d}_{s_i}, \mathbf{d}_{d_j}) - \boldsymbol{\mu}_{ij})^T \boldsymbol{\Omega}_{ij} (f(\mathbf{s}_i, \mathbf{d}_j, \mathbf{d}_{s_i}, \mathbf{d}_{d_j}) - \boldsymbol{\mu}_{ij}) \\ &\approx \sum_{\langle i,j \rangle \in \mathcal{C}} (\mathbf{r}_{ij} + \mathbf{F}_{ij}^i \Delta \mathbf{s}_i + \mathbf{F}_{ij}^j \Delta \mathbf{d}_j)^T \boldsymbol{\Omega}_{ij} (\mathbf{r}_{ij} + \mathbf{F}_{ij}^i \Delta \mathbf{s}_i + \mathbf{F}_{ij}^j \Delta \mathbf{d}_j) \end{aligned} \quad (6.26)$$

dove

$$\Delta \mathbf{s}_i = \mathbf{s}_i - \hat{\mathbf{s}}_i \quad (6.27)$$

$$\Delta \mathbf{d}_j = \mathbf{d}_j - \hat{\mathbf{d}}_j \quad (6.28)$$

$$\mathbf{r}_{ij} = f(\hat{\mathbf{s}}_i, \hat{\mathbf{d}}_j, 0, 0) - \boldsymbol{\mu}_{ij} \quad (6.29)$$

La risoluzione della rete di vincoli viene ottenuta mediante la tecnica numerica di rilassamento di Gauss-Seidel: dato  $\mathbf{x} = [\mathbf{s}_0, \dots, \mathbf{s}_n, \mathbf{d}_0, \dots, \mathbf{d}_m]^T$  il vettore di stato composto da tutti i nodi del grafo la loglikelihood negativa linearizzata può essere approssimata da

$$\chi^2(\mathbf{s}) \approx \Delta \mathbf{x}^T \mathbf{A} \Delta \mathbf{x} - 2 \Delta \mathbf{x}^T \mathbf{b} + \mathbf{c} \quad (6.30)$$

Per ottenere la stima della likelihood di partenza (6.17) è necessario trovare il minimo di questa approssimazione tramite la soluzione del sistema lineare di equazioni  $\mathbf{A} \mathbf{x} = \mathbf{b}$ . Il metodo di Gauss-Seidel prevede che il valore di ogni variabile di stato  $\mathbf{x}_i$  sia calcolato individualmente risolvendo la singola riga  $i$ , fissato il valore della variabile  $\mathbf{x}_j$  ( $i \neq j$ ). La matrice  $\mathbf{A}$  è definita tramite i blocchi  $\mathbf{A}_{ii}$  e  $\mathbf{A}_{ij}$

$$\mathbf{A}_{ii} = \sum_{\langle i,h \rangle \in \mathcal{C}} \mathbf{B}_i (\mathbf{F}_{ih}^i)^T \boldsymbol{\Omega}_{ih} \mathbf{F}_{ih}^i \mathbf{B}_i^T + \sum_{\langle h,i \rangle \in \mathcal{C}} \mathbf{B}_i (\mathbf{F}_{hi}^i)^T \boldsymbol{\Omega}_{hi} \mathbf{F}_{hi}^i \mathbf{B}_i^T \quad (6.31)$$

$$\mathbf{A}_{ij} = \begin{cases} \mathbf{B}_i (\mathbf{F}_{ij}^i)^T \boldsymbol{\Omega}_{ij} \mathbf{F}_{ij}^j \mathbf{B}_j^T & \text{se } \langle i,j \rangle \in \mathcal{C} \\ \mathbf{B}_i (\mathbf{F}_{ji}^i)^T \boldsymbol{\Omega}_{ji} \mathbf{F}_{ji}^j \mathbf{B}_j^T & \text{se } \langle j,i \rangle \in \mathcal{C} \end{cases} \quad (6.32)$$

mentre la riga  $i$  del vettore  $\mathbf{b}$ , è definita da

$$\mathbf{b}_i = \begin{cases} \sum_{\langle i,j \rangle \in \mathcal{C}} \mathbf{B}_i (\mathbf{F}_{ij}^i)^T \boldsymbol{\Omega}_{ij} \mathbf{r}_{ij} & \text{se } \langle i,j \rangle \in \mathcal{C} \\ \sum_{\langle j,i \rangle \in \mathcal{C}} \mathbf{B}_i (\mathbf{F}_{ji}^i)^T \boldsymbol{\Omega}_{ji} \mathbf{r}_{ji} & \text{se } \langle j,i \rangle \in \mathcal{C} \end{cases}$$

dove  $\mathbf{B}_i$  e  $\mathbf{F}_{ij}^i$  rappresentano la matrice di selezione e lo jacobiano della funzione di vincolo rispetto al nodo  $i$ . La notazione  $\langle i,j \rangle$  denota un vincolo tra i nodi  $i$  e  $j$ .

La soluzione dell'equazione  $i$ -esima del sistema al passo  $k$  è

$$\Delta \mathbf{x}_i^{(k)} = \mathbf{A}_{ii}^{-1} (\mathbf{b}_i - \sum_{j < i} \mathbf{A}_{ij} \mathbf{x}_j^{(k)} - \sum_{j > i} \mathbf{A}_{ij} \mathbf{x}_j^{(k-1)}) \quad (6.33)$$

La soluzione proposta permette di migliorare la precisione delle relazioni tra i nodi del grafo in modo da ottenere una rappresentazione coerente e consistente.

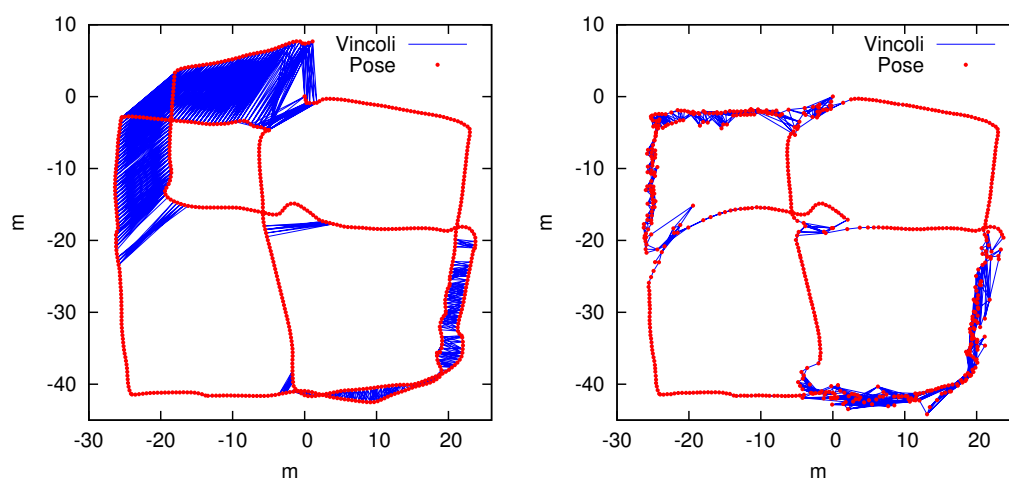


Figura 6.9: Esempio di risoluzione del grafo associato alla mappa Aces, di figura 1.4: a sinistra il grafo originale, a destra il grafo ottenuto dopo una iterazione del ciclo di risoluzione.

La figura 6.9 mostra il grafo associato ad una mappa prima e dopo l'esecuzione della procedura di risoluzione. Si può notare come le relazioni di dipendenza tra le pose (vincoli in blu) siano migliorate, e si ottenga una rappresentazione molto più simile alla mappa originale di figura 1.4.

# Conclusioni

In questa tesi è stato presentato l'insieme degli obiettivi, dei requisiti, delle soluzioni progettuali e dei risultati ottenuti in riferimento al problema della individuazione di oggetti quasi-statici da parte di un LGV dotato di laser scanner. Il lavoro svolto permette di simulare e di riprodurre mediante un robot reale la navigazione di un LGV, tramite la pianificazione e l'inseguimento di traiettorie, e di costruire una mappa degli oggetti rilevati e residenti nelle aree programmate sui sensori laser.

La soluzione proposta consiste nell'adozione di mappe locali di poligoni, una per ogni segmento in cui è divisa la traiettoria dell'LGV. Queste mappe rappresentano gli oggetti statici, quasi-statici e dinamici presenti nell'ambiente. Tale rappresentazione risulta compatta, adatta alla condivisione con altri sistemi e non aggiunge significativo carico computazionale all'applicazione. Il confronto fra le mappe locali nei diversi transiti degli LGV permette di identificare le componenti statiche o quasi-statiche della scena.

Questa soluzione si è dimostrata efficace nelle sperimentazioni che è stato possibile condurre in laboratorio e riteniamo possa essere concretamente realizzata anche nell'ambito dei sistemi industriali.

Gli sviluppi futuri di questo progetto possono riguardare:

- l'utilizzo di un riferimento assoluto più preciso di quello offerto dall'algoritmo di triangolazione utilizzato, per ottenere risultati migliori; inoltre un metodo di registrazione di scansioni (*scan matching*) potrebbe migliorare la precisione delle associazioni e quindi l'affidabilità generale dell'algoritmo;

- 
- il miglioramento del sistema di localizzazione e inseguimento traiettoria, per riuscire a simulare più segmenti, in cui ognuno mantiene una rappresentazione locale dell'ambiente osservato;
  - il miglioramento degli algoritmi di associazione, fusione e unione di poligoni, per trattare anche casi particolari che possono accadere nel funzionamento dell'applicazione, come la presenza di oggetti dinamici che possono ostruire la visuale;
  - l'esecuzione di ulteriori test in ambienti di dimensione maggiore, tramite LGV industriali equipaggiati con più di un sensore laser, per evidenziare le problematiche di fusione sensoriale e di sincronizzazione tra le rappresentazioni ottenute con diversi sensori;
  - l'introduzione nel sistema della capacità di riconoscere anche oggetti dinamici e, lavorando a livello semantico, aggiornare con intelligenza la mappa, e non puramente per via geometrica.

# Appendice A

## Configurazione aree del sensore laser

La configurazione dei sensori laser per sistemi LGV industriali prevede solitamente l'utilizzo di alcune aree di lavoro programmate sul sensore, che garantiscono il soddisfacimento dei requisiti di sicurezza del sistema. Come già descritto in 2.1, la presenza di rilevazioni nelle aree consente di individuare gli oggetti presenti nell'ambiente circostante l'LGV, e attivare le politiche di navigazione necessarie, a seconda di quale area è stata occupata. Per simulare fedelmente questa configurazione il sensore laser del robot utilizzato per questo progetto è stato programmato con due aree di sicurezza vengono descritte in seguito.

La piattaforma robotica disponibile consiste in un Pioneer P3DX [43] equipaggiato con un laser scanner Sick LMS111 [48]. Le impostazioni relative alle aree cercano di ricalcare la configurazione classica degli LGV negli impianti di stoccaggio. Il laser è stato configurato tramite il software SOPAS [49], con cui sono state create due aree di sicurezza di dimensioni opportune per permettere la navigazione agevole del robot negli ambienti di prova. Il software SOPAS permette di creare aree di diversi tipi tra cui forme poligonali che possono essere disegnate in qualunque posizione all'interno del piano di scansione. Le aree utilizzate in questo progetto sono rappresentate

da poligoni e si distinguono in:

- AREA DI PREAVVISO: è l'area più estesa che serve per segnalare ostacoli lontani quando l'LGV si muove a velocità elevate, in figura A.1. Il robot rallenta quando questa area è attiva;

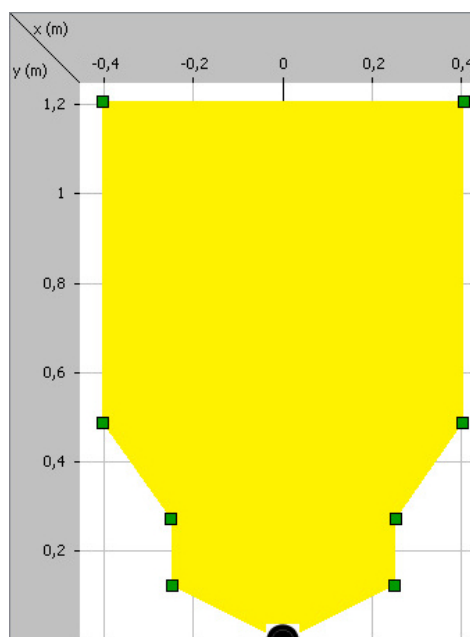


Figura A.1: Area di preavviso.

- AREA DI ALLARME: solitamente è l'area più piccola e interna che segnala ostacoli immediati, mostrata in figura A.2, e può causare la fermata del robot se attivata, per garantire l'incolumità di persone e cose nelle vicinanze.

Il laser scanner dispone di una interfaccia ethernet e di tre segnale di output digitali, il cui valore può essere utilizzato per pilotare altro hardware collegato oppure può servire per identificare in ogni momento qual'è la combinazione delle aree attiva. Il valore degli output può essere ricavato dal laser anche tramite l'interfaccia ethernet: in particolare il pacchetto dati per la comunicazione di rete prevede al suo interno un campo che viene codificato in esadecimale a 3 bit, quindi con valori compresi tra 0 (tutti gli output off)

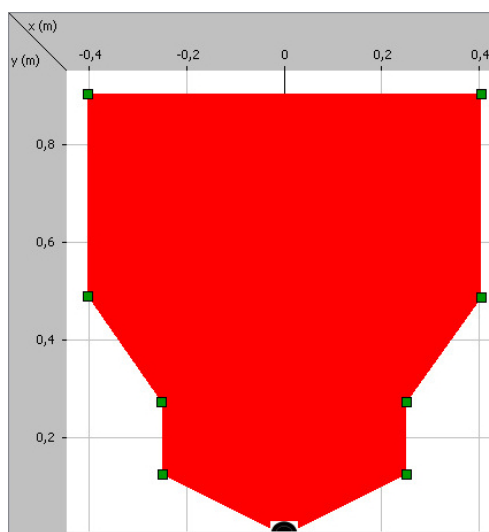


Figura A.2: Area di allarme.

e 7 (tutti gli output on), che viene usato per leggere la configurazione degli output. Questo campo è in posizione 14 nella trama del pacchetto ethernet come mostrato in figura A.3.

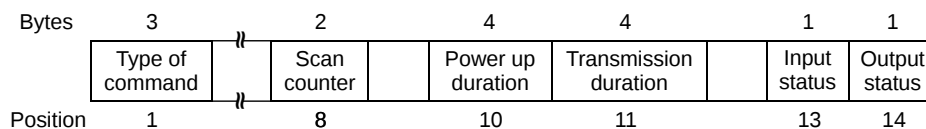


Figura A.3: Parte del formato del pacchetto dati ethernet per il laser Sick LMS111.

questi output sono associati alle configurazioni possibili di aree tramite il SOPAS.

Per intervenire sugli output si possono creare dei “casi di valutazione”, che descrivono i possibili eventi nell’attivazione delle aree. Ogni caso consiste in una o più strategie di valutazione delle aree; il laser mette a disposizione le seguenti strategie:

- *valutazione pixel*: il laser valuta l’intera area, se un oggetto entra nell’area l’evento viene inviato alla relativa uscita;
- *blanking*: il laser valuta l’intera area, ma viene settata una dimensione di blanking che viene utilizzata come filtro. Se un oggetto entra nel-

l'area ed è più largo della larghezza di blanking viene riconosciuto, e il risultato viene inviato agli output;

- *contorni*: viene valutata la presenza di un contorno che deve rimanere in modo permanente nell'area;
- *operator I/O*: un operatore che permette di mappare gli input del laser sugli output.

Quando una di queste strategie diventa attiva, l'area del caso di valutazione corrispondente viene segnalata come occupata, e di conseguenza anche gli output associati diventano attivi.

In questo lavoro sono stati definiti due casi di valutazione, e per ognuno di essi è stata creata un'area che attiva un bit in output quando è occupata, tramite la strategia *valutazione pixel*, cioè nel momento in cui un beam ricade nel poligono descritto dall'area stessa. Sono sufficienti due bit per mappare le tre configurazioni disponibili per le aree: area di preavviso attiva, area di allarme attiva oppure entrambe attive.

# Appendice B

## Jacobiani della funzione di vincolo

In questa sezione vengono descritte le operazioni di composizione e decomposizione utili per formalizzazione della rete di vincoli del grafo, utilizzato come supporto per il modello SPmap presentato nel capitolo 6, nell'ambito del Graph-SLAM. La composizione tra due vettori  $\mathbf{s}$  e  $\mathbf{d}$  è definita da:

$$c(\mathbf{s}, \mathbf{d}) = \mathbf{s} \oplus \mathbf{d} = \begin{bmatrix} s_x + d_x \cos s_\theta - d_y \sin s_\theta \\ s_y + d_x \sin s_\theta + d_y \cos s_\theta \\ s_\theta + d_\theta \end{bmatrix} \quad (\text{B.1})$$

Gli jacobiani della composizione rispetto alla prima e alla seconda variabile sono:

$$\mathbf{C}_{sd}^s(\mathbf{s}, \mathbf{d}) = \begin{bmatrix} 1 & 0 & -d_x \sin s_\theta - d_y \cos s_\theta \\ 0 & 1 & d_x \cos s_\theta - d_y \sin s_\theta \\ 0 & 0 & 1 \end{bmatrix} \quad (\text{B.2})$$

$$\mathbf{C}_{sd}^d(\mathbf{s}, \mathbf{d}) = \begin{bmatrix} \cos s_\theta & -\sin s_\theta & 0 \\ \sin s_\theta & \cos s_\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (\text{B.3})$$

La decomposizione tra due vettori  $\mathbf{s}$  e  $\mathbf{d}$  è definita da:

$$d(\mathbf{s}, \mathbf{d}) = \mathbf{s} \ominus \mathbf{d} = \begin{bmatrix} (s_x - d_x) \cos d_\theta + (s_y - d_y) \sin d_\theta \\ -(s_x - d_x) \sin d_\theta + (s_y - d_y) \cos d_\theta \\ s_\theta - d_\theta \end{bmatrix} \quad (\text{B.4})$$

Gli jacobiani della decomposizione rispetto alla prima e alla seconda variabile sono:

$$\mathbf{D}_{sd}^S(\mathbf{s}, \mathbf{d}) = \begin{bmatrix} \cos d_\theta & \sin d_\theta & 0 \\ -\sin d_\theta & \cos d_\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (\text{B.5})$$

$$\mathbf{D}_{sd}^d(\mathbf{s}, \mathbf{d}) = \begin{bmatrix} -\cos d_\theta & -\sin d_\theta & -(s_x - d_x) \sin d_\theta + (s_y - d_y) \cos d_\theta \\ \sin d_\theta & -\cos d_\theta & -(s_x - d_x) \cos d_\theta - (s_y - d_y) \sin d_\theta \\ 0 & 0 & -1 \end{bmatrix} \quad (\text{B.6})$$

Dato un vertice qualunque del grafo che connette due nodi,  $\mathbf{s}$  (sorgente del vertice), e  $\mathbf{d}$  (destinazione del vertice), la funzione del vincolo associato al vertice tra i due nodi del grafo risulta essere:

$$f(\mathbf{s}, \mathbf{d}, \mathbf{d}_s, \mathbf{d}_d) = (\mathbf{s} \oplus \mathbf{d}_s) \ominus (\mathbf{d} \oplus \mathbf{d}_d) = d(c(\mathbf{s}, \mathbf{d}_s), c(\mathbf{d}, \mathbf{d}_d)) \quad (\text{B.7})$$

Gli jacobiani della funzione di vincolo sono utilizzati per la linearizzazione di  $f$  tramite espansione di Taylor del primo ordine attorno al punto  $(\hat{\mathbf{s}}, \hat{\mathbf{d}}, 0, 0)$ , e sono:

$$\mathbf{F}_{sd}^s(\mathbf{s}, \mathbf{d}, \mathbf{d}_s, \mathbf{d}_d)|_{\mathbf{d}_s=0, \mathbf{d}_d=0} = \mathbf{D}_{sd}^d(\mathbf{s}, \mathbf{d}) \mathbf{C}_{sd}^d(\mathbf{d}, 0) = \quad (\text{B.8})$$

$$= \begin{bmatrix} -1 & 0 & (s_x - d_x) \sin s_\theta + (d_y - s_y) \cos s_\theta \\ 0 & -1 & (s_y - d_y) \sin s_\theta + (s_x - d_x) \cos s_\theta \\ 0 & 0 & -1 \end{bmatrix} \quad (\text{B.9})$$

$$\mathbf{F}_{sd}^d(\mathbf{s}, \mathbf{d}, \mathbf{d}_s, \mathbf{d}_d)|_{\mathbf{d}_s=0, \mathbf{d}_d=0} = \mathbf{D}_{sd}^s(\mathbf{s}, \mathbf{d}) \mathbf{C}_{sd}^d(\mathbf{s}, 0) = \quad (\text{B.10})$$

$$= \begin{bmatrix} \cos d_\theta \cos s_\theta + \sin d_\theta \sin s_\theta & -\sin d_\theta \cos s_\theta + \cos d_\theta \sin s_\theta & 0 \\ -\cos d_\theta \sin s_\theta + \sin d_\theta \cos s_\theta & \cos d_\theta \cos s_\theta + \sin d_\theta \sin s_\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (\text{B.11})$$

# Bibliografia

- [1] H. Moravec and A. Elfes. “High resolution maps from wide angle sonar”. In Proc. ICRA, pages 116-121, 1985.
- [2] W. Burgard, A. Cremers, D. Fox, G. Lakemeyer, D. Hahnel, D. Schulz, W. Steiner, and S. Thrun. “The interactive museum tour-guide robot”. In Proc. AAAI, 1998.
- [3] S. Thrun, D. Fox, and W. Burgard, “A probabilistic approach to concurrent mapping and localization for mobile robots”. *Mach. Learning Autom. Robots*, vol. 31, pp. 29-53, 1998.
- [4] L. J. Latecki, R. Lakaemper, X. Sun and D. Wolter “Building polygonal maps from laser range data”. In Proc. Int. Cognitive Robotics Workshop, p.56 , 2004.
- [5] F. Amigoni, S. Gasparini, and M. Gini, “Building segment-based maps without pose information”. *Proceedings of the IEEE*, vol. 94, no. 7, pp. 1340-1359, 2006.
- [6] E. Arkin, L. P. Chew, D. P. Huttenlocher, K. Kedem, and J. S. B. Mitchell, “An efficiently computable metric for comparing polygonal shapes”. *IEEE Trans. Patt. Anal. Machine Intell.*, vol. 13, no. 3, pp. 209-216, 1991.
- [7] F. Amigoni and S. Gasparini, “Analysis of methods for reducing line segments in maps: Towards a general approach”. In Proc. IEEE/RSJ IROS 2008, pp. 2896-2901.

- 
- [8] R. Sim, G. Dudek, "Learning generative models of scene features". In Proc. of IEEE Intl Conf. Computer Vision and Pattern Recognition (CVPR), vol. 1, pp. 406-412 (2001).
- [9] R. Sim, G. Dudek, "Effective exploration strategies for the construction of visual maps". In Proc. of 5th IFAC/EURON Symposium on Intelligent Autonomous Vehicles, vol. 3, pp. 3224-3231 (2003).
- [10] R. Sim., P. Elinas, M. Griffin, A. Shyr, J.J. Little, "Design and analysis of a framework for real-time vision-Based SLAM using Rao-Blackwellised particle filters". In Proc. of the 3rd Canadian Conference on Computer and Robotic Vision (2006).
- [11] N. Winters, J. Santos-Victor, "Omnidirectional visual navigation". In Proc. of IEEE Int'l Symposium on Intelligent Robotic Systems (SIRS), pp. 109-118 (1999).
- [12] A. Remazeilles, F. Chaumette, P. Gros, "3D navigation based on a visual memory". In Proc. of IEEE Int'l Conf. on Robotics and Automations (ICRA), pp. 2719-2725 (2006).
- [13] S. Se, D.G. Lowe, J.J. Little, "Vision-based global localization and mapping for mobile robots". Robotics, IEEE Transactions on vol.21, no.3, pp. 364-375, June 2005.
- [14] D. G. Lowe, "Object recognition from local scale-invariant features". In Proc. 7th Int. Conf. Computer Vision (ICCV 99), Kerkyra, Greece, Sep.1999, pp. 1150-1157.
- [15] M. Cummins, P. Newman, "Probabilistic Appearance Based Navigation and Loop Closing". 2007 IEEE International Conference on Robotics and Automation, pp.2042-2048, 10-14 April 2007.
- [16] FLIRT, features multi scala per sensori laser o sonar,  
<http://www.openslam.org/flirtlib.html>

- 
- [17] J. Leonard and H. Durrant-Whyte, "Simultaneous map building and Localization for an Autonomous Mobile Robot". In Proc. IEEE Int. Wkshp. Intell. Robots Syst., pp.1442-1447.
- [18] G. Grisetti, C. Stachniss, and W. Burgard. "Improved techniques for grid mapping with rao-blackwellized particle filters". IEEE Transactions on Robotics, 23(1):34-46, 2007.
- [19] S. Thrun, Y. Liu, D. Koller, A.Y. Ng, Z. Ghahramani, and H. Durrant-Whyte. "Simultaneous localization and mapping with sparse extended information filters". Int. Journal of Robotics Research, 23(7/8):693-716, 2004.
- [20] F. Lu and E. Milios. "Globally consistent range scan alignment for environment mapping". Autonomous Robots, 4:333-349, 1997.
- [21] F. Dellaert and M. Kaess. "Square root SAM: Simultaneous location and mapping via square root information smoothing". Int. Journal of Robotics Research, 2006.
- [22] A. Howard, M.J. Mataric, and G. Sukhatme. "Relaxation on a mesh: a formalism for generalized localization". In Proc. of the IEEE/RSJ IROS, 2001.
- [23] U. Frese, P. Larsson, and T. Duckett. "A multilevel relaxation algorithm for simultaneous localisation and mapping". IEEE Transactions on Robotics, 21(2):1-12, 2005.
- [24] E. Olson, J. Leonard, and S. Teller. "Fast iterative optimization of pose graphs with poor initial estimates". In Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA), pp. 2262-2269, 2006.
- [25] S. Thrun and M. Montemerlo. "The graph SLAM algorithm with applications to large-scale mapping of urban structures". Int. Journal of Robotics Research, 25(5-6):403, 2006.

- 
- [26] J. D. Tardos, “Representing partial and uncertain sensorial information using the theory of symmetries”. In Proc. 1992 IEEE Int. Conf. Robot. Automat., Nice, France, 1992, pp. 1799-1804.
- [27] Jose A. Castellanos, J. M. M. Montiel, J. Neira, and J. D. Tardos. “The SPMAP: A Probabilistic Framework for Simultaneous Localization and Map Building”. IEEE Transactions on Robotics & Automation 15(5):948-953, 1999.
- [28] L. J. Latecki, R. Lakamper, “Shape similarity measure based on correspondence of visual parts”, Pattern Analysis and Machine Intelligence, IEEE Transactions on, On page(s): 1185-1190, Volume: 22 Issue: 10, Oct 2000.
- [29] L.J. Latecki, R. Lakamper, “Convexity Rule for Shape Decomposition Based on Discrete Contour Evolution”, Computer Vision and Image Understanding, vol. 73, pp. 441-454, 1999.
- [30] M. Arkin, L.P. Chew, D.P. Huttenlocher, K. Kedem, and J.S.B. Mitchell, “An Efficiently Computable Metric for Comparing Polygonal Shapes”. IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 13, pp. 209-206, 1991.
- [31] L. Montesano, J. Minguez and L. Montano, “Modeling the static and the dynamic parts of the environment to improve sensor-based navigation”, in Proc. IEEE Int. Conf. on Robotics & Automation (ICRA), p. 4556, 2005.
- [32] G. Tipaldi, F. Ramos, “Motion clustering and estimation with conditional random fields”. In Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2009.
- [33] J. Lafferty, A. McCallum, and F. Pereira. “Conditional random fields: Probabilistic models for segmenting and labeling sequence data”. In Proc. of the Int. Conf. on Machine Learning (ICML), 2001.

- 
- [34] W. Burgard, C. Stachniss, and D. Haehnel. “Mobile robot map learning from range data in dynamic environments”. In *Autonomous Navigation in Dynamic Environments*, volume 35 of *Springer Tracts in Advanced Robotics*. Springer Verlag, 2007.
- [35] D. Meyer-Delius, J. Hess, G. Grisetti, and W. Burgard. “Temporary maps for robust localization in semi-static environments”. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, Taipei, Taiwan, 2010.
- [36] G. Grisetti, C. Stachniss, and W. Burgard, “Improving grid-based slam with rao-blackwellized particle filters by adaptive proposals and selective resampling”. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2005.
- [37] G. Grisetti, R. Kummerle, C. Stachniss, U. Frese, and C. Hertzberg, “Hierarchical optimization on manifolds for online 2d and 3d mapping”. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2010.
- [38] J. S. Esteves, A. Carvalho, and C. Couto, “Generalized geometric triangulation algorithm for mobile robot absolute self-localization”. In *IEEE International Symposium On Industrial Electronics (ISIE)*, Rio de Janeiro, 2003.
- [39] Cohen, Charles and Koss, Frank V., “A Comprehensive Study of Three Object Triangulation”. *Mobile Robots VII*, SPIE Vol. 1831, 1992.
- [40] Josep M. Font Llagunes, Joaquim A. Batlle, “Consistent triangulation for mobile robot localization using discontinuous angular measurements”. *Robotics and Autonomous Systems* 57 (2009) pp. 931-942, 2009.
- [41] D. Ronzoni, R. Olmi, C. Secchi and C. Fantuzzi “AGV Global Localization Using Indistinguishable Artificial Landmarks”. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2011.

- 
- [42] D. H. Douglas and T. K. Peucker. “Algorithms for the reduction of the number of points required to represent a line or its caricature”. *The Canadian Cartographer*, 10(2):112-122, 1973.
- [43] Pioneer 3-DX, MobileRobots, <http://www.mobilerobots.com/researchrobots/researchrobots/pioneer3dx.aspx>
- [44] Sebastian Thrun, Wolfram Burgard, Dieter Fox, “Probabilistic Robotics”. *Intelligent Robotics and Autonomous Agents series*, Chap. 3.3, pp. 54-61, MIT, 2005.
- [45] Bruno Siciliano, Khatib Oussama, “Handbook of Robotics”. Part E, Chap. 34.4.1-34.4.2, pp. 807-810, Springer, 2008.
- [46] D. Hahnel, W. Burgard, B. Wegbreit, and S. Thrun. “Towards lazy data association in slam”. In *Proc. of the Int. Symposium of Robotics Research (ISRR)*, pages 421-431, Siena, Italy, 2003.
- [47] J. Neira and J.D. Tardos. “Data association in stochastic mapping using the joint compatibility test”. *IEEE Transactions on Robotics and Automation*, 17(6):890-897, 2001.
- [48] Sick laser scanner LMS111.
- [49] Sick SOPAS, tool di configurazione per laser scanner.