



UNIVERSITA' DEGLI STUDI DI ROMA  
TOR VERGATA

FACOLTA' DI INGEGNERIA

CORSO DI LAUREA IN INGEGNERIA  
DELL' AUTOMAZIONE

A.A. 2009/2010

*Tesi di Laurea*

*Algoritmi per la costruzione di mappe  
per la navigazione di robot mobili*

RELATORE

Ing. Francesco Martinelli

CANDIDATO

Gabriele Petrini



# Indice

<b>1. Introduzione</b>	<b>1</b>
<b>2. Problematiche Generali della Robotica Mobile</b>	<b>4</b>
2.1 Introduzione	4
2.2 Rappresentazione dell'informazione	6
2.2.1 Mappe Metriche	7
2.2.2 Mappe Topologiche e Ibride	9
2.3 Stima Stocastica dello Stato	10
2.3.1 Filtro di Bayes	11
2.3.2 Modello del moto	15
2.3.3 Modello del sensore	17
2.4 Filtri di Bayes Approssimati	21
2.4.1 Filtro Binario di Bayes	23

<b>3. Tecniche di Mappatura</b>	<b>25</b>
3.1 La Griglia Occupazionale	25
3.2 Mappatura basata su Filtro Binario	29
3.3 Mappatura basata su Stimatore MAP	35
3.3.1 Costruzione dello Stimatore MAP	37
3.3.2 Algoritmi “Hill Climbing”	41
3.3.3 Implementazione dell’algoritmo	42
3.4 Tecniche Alternative	45
3.4.1 Mappatura basata sulla probabilità di Riflessione	45
3.4.2 Mappatura basata sulla probabilità di Copertura	46
<b>4. Simulazioni</b>	<b>47</b>
4.1 Introduzione	47
4.2 L’ambiente, il robot, il sensore	48
4.3 Simulazioni	50
4.4 Conclusioni e Sviluppi futuri	56
<b>Bibliografia</b>	<b>58</b>





# Capitolo 1

## Introduzione

Nel corso del tempo grazie allo sviluppo di tecnologie sempre più accurate e affidabili si è affiancato al lavoro dell'uomo quello del robot. Un robot è un insieme di dispositivi che interagiscono con l'ambiente circostante (attuatori) e ottengono misurazioni (sensori) al fine di raggiungere un obiettivo. Affinché sia un valido supporto al genere umano il robot deve acquisire un certo grado di autonomia, deve essere in grado cioè di poter ricevere segnali dall'esterno, elaborarli e prendere la giusta decisione. Se pensiamo a siti post disastro potenzialmente pericolosi per l'uomo, sarebbe preferibile l'invio di robot in grado di partire da un punto, esplorare l'intero sito, costruire una mappa dettagliata e ritornare al punto di partenza. La disciplina che si occupa dello studio delle metodologie che permettono la navigazione autonoma di un robot è la Robotica Mobile. Negli anni tale scienza si è sviluppata notevolmente portando i suoi risultati in molti campi applicativi: dal mini-robot esploratore, ai manipolatori e sistemi di trasporto usati nell'industria, dalla guida assistita e parcheggio assistito delle automobili, fino all'esplorazione di interi pianeti. Per quanto riguarda il contributo che

robot mobili possono dare all'uomo molti progetti sono stati fatti o sono in fase di esecuzione. A titolo di esempio citiamo i due prototipi messi a punto presso l'Istituto di Scienze e Tecnologie della Cognizione (ISTC) del CNR: il RobotCare[3] e il RobotCup Rescue[4]. Il primo è un progetto che mira ad utilizzare robot mobili per l'assistenza degli anziani. Il secondo si interessa di promuovere la ricerca nel settore dell'organizzazione di



Figura 1.1 Progetto RobotCare [3]

squadre di robot autonomi per il soccorso di popolazioni colpite da disastri naturali. Per espletare questi compiti il robot deve essere in grado di risolvere l'enorme incertezza del mondo fisico. I fattori che concorrono all'aumento dell'incertezza sono molteplici: presenza di persone nelle vicinanze del robot, limitazione del raggio di azione dei sensori fino ad un range massimo, rumori associati ad ogni lettura del sensore, approssimazione degli algoritmi, e così via. Nel corso degli anni l'approccio probabilistico si è rivelato essere il migliore in quanto meglio rappresenta l'incertezza. Infatti grazie a tale metodo i problemi di localizzazione, SLAM e mapping vengono ricondotti al problema della stima dello stato, attraverso l'uso di un algoritmo iterativo (*Fil-*

tro di Bayes). La rappresentazione dell'informazione viene modellata con una funzione densità di probabilità e l'aggiornamento di tale funzione equivale all'informazione acquisita con le misure. Il problema della robotica mobile non si presenta come un unico blocco, ma come l'insieme di vari sottoproblemi. Ad esempio potrei essere interessato, avendo a disposizione a priori una mappa dell'ambiente, ad ottenere la posizione del robot in un particolare momento (localizzazione passiva); in altre situazioni invece non è disponibile una mappa e l'unica soluzione è quella di esplorare l'ambiente costruendone una e *contemporaneamente* individuare la posizione all'interno di tale mappa (SLAM). Nel corso di questo lavoro ci occuperemo di un altro importante sottoproblema riguardante la robotica mobile: il Mapping. L'obiettivo del mapping è quello di costruire una rappresentazione dell'informazione spaziale (una mappa) partendo da un insieme di dati incerti e rumorosi ottenuti dalle misurazioni. Il Mapping prevede quindi la conoscenza a priori delle posizioni assunte dal robot ed è per questo che ci si riferisce ad esso con il nome di Mapping con posizioni note (Mapping with known poses). Questo lavoro tratterà nella prima parte problematiche generali della robotica mobile e i mezzi usati per risolverle, nella seconda parte presenteremo un algoritmo di stima massima a posteriori (Maximum A Posteriori Estimator) con il quale risolveremo il problema del Mapping. Tale metodo sarà poi confrontato con un'altra tecnica di mappatura, evidenziandone i miglioramenti e le limitazioni. L'approccio usato per la realizzazione di tale tesi è stato prettamente teorico-simulativo: la validità degli algoritmi presentati è stata verificata mediante simulazioni.



# Capitolo 2

## Problematiche generali della robotica mobile

### 2.1 Introduzione

La navigazione autonoma di un robot è legata al raggiungimento di diversi obiettivi. Un compito potrebbe essere quello di descrivere l'ambiente circostante durante l'avanzamento lungo un percorso; oppure, avendo a disposizione una rappresentazione dell'ambiente, individuare la nostra posizione al suo interno; oppure, conoscendo la mappa e la posizione di partenza, scegliere il percorso “migliore” per raggiungere una destinazione. Tutti questi problemi rappresentano i pilastri della robotica mobile e vengono formalizzati con il nome di: *Mapping*, *Localizzazione* e *Pianificazione & Controllo*. Come si può immaginare, ognuno di essi non è indipendente, ma fortemente legato all'altro. Il Mapping si occupa di costruire una mappa dello spazio operativo avendo a disposizione le misure effettuate dai sensori (i.e. richiede la conoscenza a priori delle posizioni assunte dal robot). La Localizzazione è il problema di determina-

re la posizione del robot avendo a disposizione una mappa. Mapping e Localizzazione sono legati tra di loro e sembra impossibile risolvere uno senza risolvere prima l'altro. Ecco perché si sono sviluppate tecniche di Mappatura e Localizzazione Simultanea (SLAM). La localizzazione è formata da molti sottoproblemi. Parliamo di *Localizzazione Locale* (tracciamento della posizione, *Position Tracking*) quando la posizione iniziale è nota. Parliamo di *Localizzazione globale* quando anche la posizione iniziale è sconosciuta. Una variante al problema di quest'ultima è il cosiddetto *Kidnapped Robot Problem*: durante il suo funzionamento il robot viene prelevato e trasportato a sua insaputa in una nuova posizione, aumentando così il grado di difficoltà del problema.

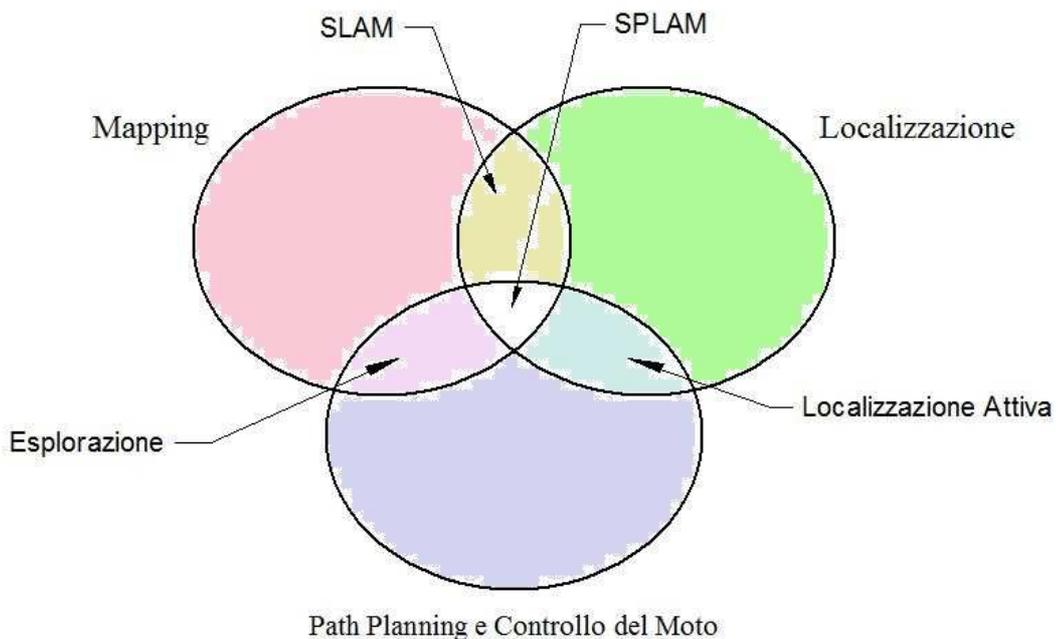


Figura 2.1 Caratteristiche della Robotica Mobile

L'ultima grande classe di funzioni inerenti alla robotica mobile è caratterizzata dal *Path Planning* e *Controllo del moto* (in genere queste due definizioni si equivalgono in quanto il path planning è effettuato attraverso un controllo del moto). Il path

planning si occupa di pianificare il percorso, il controllo del moto di impartire i comandi opportuni. Di solito durante la localizzazione i segnali di controllo provengono dall'esterno (*Localizzazione Passiva*). Si può pensare di generare i segnali di controllo direttamente dal robot in modo da minimizzare il costo della localizzazione (*Localizzazione Attiva*). Ne è un esempio la navigazione costiera (*costal navigation*) dove il robot sceglie di seguire un percorso vicino alle pareti dell'ambiente per facilitare la localizzazione. Un'ulteriore applicazione delle tecniche di path planning è quello di muoversi in un ambiente sconosciuto privi di mappa per massimizzare la conoscenza del mondo esterno (*Esplorazione*). La forte dipendenza tra i vari sottoproblemi ha favorito la nascita di approcci integrati grazie ai quali il robot può contemporaneamente costruire una mappa, localizzarsi e generare i controlli appropriati (SPLAM, Simultaneous Planning Localization And Mapping). Nella figura 2.1 vengono mostrate le componenti della navigazione autonoma.

## 2.2 Rappresentazione dell'informazione

Lo scopo di questa tesi è quello di presentare algoritmi che siano in grado di generare rappresentazioni dell'ambiente che si discostino il meno possibile dall'ambiente reale. I modi in cui una mappa può rappresentare il mondo reale sono molteplici e la scelta di uno rispetto all'altro dipende fortemente dalla struttura dello spazio operativo. In generale le tipologie di mappe possono essere divise in tre categorie:

- mappe metriche
- mappe topologiche
- mappe ibride

## 2.2.1 Mappe metriche

Le rappresentazioni metriche si basano sulla descrizione di caratteristiche geometriche dell'ambiente. Una mappa può essere vista come una lista di oggetti ognuno dei quali memorizza una caratteristica dell'ambiente e la sua posizione. Una mappa può quindi apparire nella forma:

$$m = \{m_1, m_2, \dots, m_N\}$$

dove  $N$  è il numero totale di oggetti nell'ambiente e ogni  $m_i$  specifica una proprietà.

All'interno delle mappe metriche troviamo due modelli:

- feature-based map
- location-based map

Le mappe basate sull'aspetto dell'ambiente (feature-based map) descrivono lo spazio operativo come l'insieme di elementi caratteristici (landmark). Una tale mappa sarà dunque una lista in cui ogni  $m_i$  indica la forma geometrica dell'oggetto e la sua posizione nello spazio. Se gli oggetti di base utilizzati per descrivere la mappa sono adatti a rappresentare le caratteristiche presenti nell'ambiente, tali mappe permettono di ottenere una descrizione accurata. Esse possono essere più compatte rispetto ad altri tipi (ad esempio Occupancy Grid) in quanto rappresentano solo gli oggetti presenti e non punti privi di qualsiasi interesse. Inoltre si prestano meglio a descrivere ambienti dinamici in cui la posizione degli oggetti può cambiare nel tempo. Infine esse sono più vicine alla percezione che l'uomo ha dell'ambiente facilitando l'interazione con la macchina. Tali mappe hanno sostanzialmente tre svantaggi: sono limitate agli ambienti descrivibili dalle forme geometriche di base perdendo tutta l'informazione che sensori moderni possono fornire; soffrono del cosiddetto *Data Association Problem* ovvero capire di aver osservato lo stesso oggetto in due istanti di tempo diversi (problema dei cicli); infine per poter estrapolare forme geometriche dallo spazio bisogna munirsi

di sensori adeguati non banali.

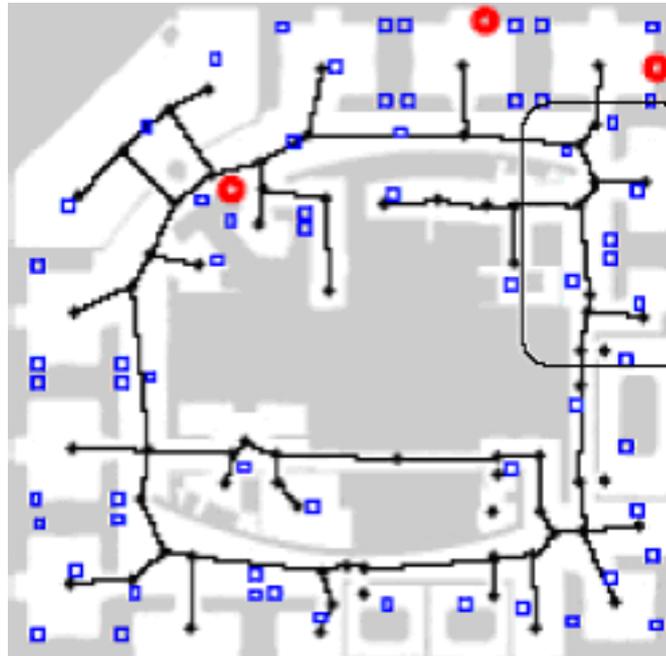


Figura 2.2 Mappa basata su landmark [7]

Le mappe basate sulla posizione (location-based map) forniscono un'etichetta per ogni luogo. In esse ogni  $m_i$  contiene le coordinate della posizione e un valore che rappresenta una proprietà di quel luogo. Per come sono state definite oltre a dare informazioni sulla posizione degli oggetti indicano anche gli spazi vuoti. Nel corso di questo lavoro faremo uso di una delle più popolari forme di location-based map: la *Mappa basata su griglie di occupazione*. Tale mappa divide lo spazio di lavoro (assunto bi-dimensionale) in un numero prestabilito di celle e ad ogni cella viene associato un valore numerico che ne indica lo stato di occupazione. A differenza delle mappe basate su Landmark che ben rappresentano spazi strutturati, queste mappe sono adatte a rappresentare qualsiasi tipo di ambiente. La precisione della mappa può essere regolata attraverso la dimensione assegnata alla singola cella e si può far in modo che la dimensione delle celle sia variabile e non fissa in modo tale che in zone dell'ambiente

ricche di caratteristiche le celle assumano una dimensione minore per aumentare il grado di accuratezza della mappa. Sfortunatamente tali mappe non possono essere usate per la rappresentazione di ambienti dinamici in quanto richiedono l'assunzione di mondo statico. Tali mappe si prestano bene a risolvere il problema dell'esplorazione, in quanto ben rappresentano gli spazi vuoti. Una volta ottenuta la mappa, si può pensare di realizzare un grafo i cui nodi rappresentano le celle o i gruppi di celle adiacenti libere ed esiste un arco di collegamento tra i diversi nodi se esiste almeno una cella libera che li unisce. In questo modo il problema dell'esplorazione si riduce al calcolo di un cammino minimo (se esiste) tra un nodo sorgente e una destinazione (ad esempio usando l'algoritmo di Dijkstra). Tale metodo di esplorazione è sicuramente molto più semplice da implementare rispetto ad altri che si basano ad esempio su campi di potenziali.

## 2.2.2 Mappe Topologiche e Ibride

Le mappe topologiche rappresentano l'ambiente attraverso un grafo. Tali mappe sono una rappresentazione più compatta e vengono usate per fare ragionamenti ad alto livello per risolvere problemi quali l'esplorazione, la pianificazione di percorsi e così via. Generalmente l'approccio usato è quello descritto sopra: dopo la generazione della mappa metrica si genera quella topologica. In questo caso si parla di mappe ibride, intendendo l'uso contemporaneo delle due tipologie.

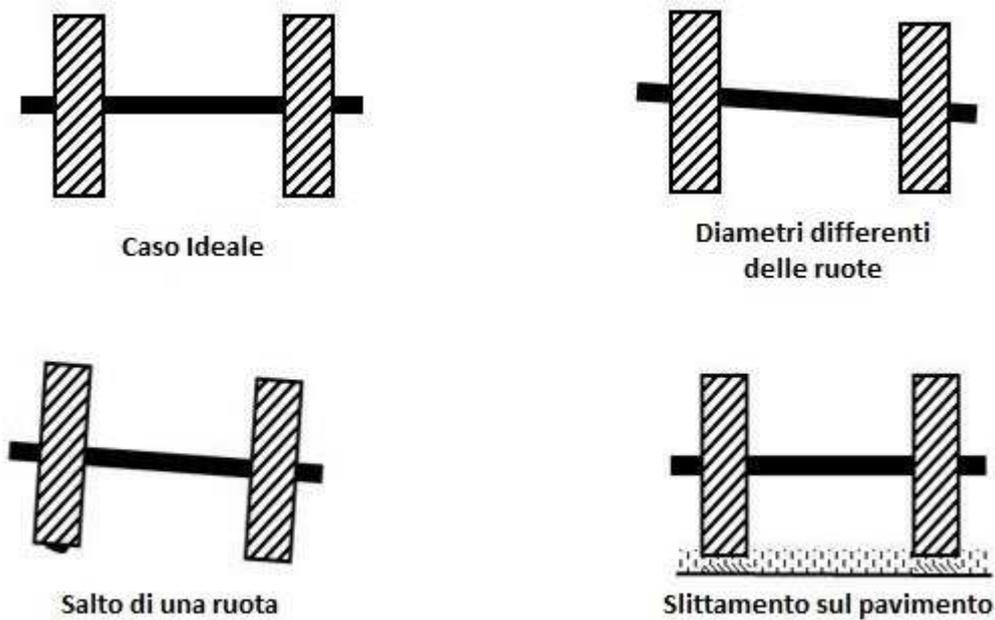


Figura 2.3 Cause di rumore [1]

## 2.3 Stima Stocastica dello Stato

Il problema della robotica mobile è quello di poter determinare lo stato del sistema al tempo  $t$  ( $x_t$  = posizione, mappa o entrambi) partendo da una serie di misure incerte e rumorose. Un metodo che ci permette di determinare la stima dello stato di un sistema è l'uso degli osservatori o stimatori. L'osservatore è un sistema dinamico usato per determinare grandezze non misurabili direttamente (o quando l'errore di misura sarebbe più grande della stima stessa), il cui valore deve essere ricavato attraverso la misura di altre grandezze e l'utilizzo di un modello matematico. Esistono due tipi di osservatore:

- osservatore deterministico (o di Luenberger), caratterizzato dal fatto che il modello che rappresenta il sistema da osservare è perfetto e le misure sono prive di errori.
- osservatore stocastico, che considera rumorosi i segnali che riceve in ingresso.

Le sorgenti di rumore sono molteplici (imprecisione dei sensori, imperfezioni dell'ambiente, ecc) e rendono impossibile l'uso di osservatori deterministici. Nella figura 2.3 si evidenziano alcune cause di errori sistematici e accidentali.

Uno strumento matematico usato per la costruzione di stimatori stocastici dello stato è il *Filtro di Bayes*.

### 2.3.1 Filtro di Bayes

Il filtro di Bayes è un'applicazione ricorsiva della *formula di Bayes*, che aggiorna progressivamente la stima dello stato ad ogni arrivo di una nuova misura. La formula di Bayes:

$$p(x_i | z) = \frac{p(z | x_i)p(x_i)}{p(z)}$$

afferma che la probabilità che la variabile di stato  $x_i$  assuma un determinato valore conoscendo la misura  $z$  è il prodotto di due termini  $p(z | x_i)$  e  $p(x_i)$ . Il primo rappresenta la probabilità di osservare una misura  $z$  conoscendo  $x_i$ . Esso è un modello generativo nel senso che descrive la generazione della misura conoscendo i diversi stati. Il secondo termine indica la probabilità che la variabile di stato assuma un particolare valore senza conoscere la misura  $z$  e per questo viene detta probabilità *a priori*. Di conseguenza il termine  $p(x_i | z)$  assume il ruolo di probabilità *a posteriori*. Il termine  $p(z)$  rappresenta una costante ed è per questo che viene indicato come un fattore di normalizzazione ( $\eta = p(z)^{-1}$ ), usato affinché la quantità a sinistra dell'uguaglianza

sia una funzione densità di probabilità. Possiamo scrivere così la forma compatta della regola di Bayes:

$$\text{Posterior} = \eta \cdot \text{Likelihood} \cdot \text{Prior}$$

Essendo uno stimatore stocastico, il filtro di Bayes produce la stima dello stato  $x_t$  come una funzione densità di probabilità dipendente dagli ingressi (i controlli  $u_t$ ) e dalle uscite (le misure  $z_t$ ). Tale stima prende il nome di *Belief* ( $bel(x_t)$ ). Applicando la formula di Bayes otteniamo:

$$(2.1) \quad bel(x_t) = p(x_t | z_{1:t}, u_{1:t}) = \frac{p(z_t | x_t, z_{1:t-1}, u_{1:t}) p(x_t | z_{1:t-1}, u_{1:t})}{p(z_t | z_{1:t-1}, u_{1:t})}$$

In molte applicazioni pratiche si assume l'ipotesi che il processo osservato sia *Markoviano*. Un processo si dice Markoviano quando lo stato corrente è una statistica sufficiente di tutti i controlli e le misure passate fino al suddetto istante. Quindi tale assunzione afferma che i dati passati e futuri sono indipendenti se conosciamo lo stato corrente  $x_t$ . Esistono numerosi fattori che portano alla violazione di tale ipotesi: variabili dell'ambiente non incluse nello stato  $x_t$ , imprecisioni nei modelli probabilistici, errori di approssimazione. La soluzione potrebbe essere quella di includere tutte le variabili non modellate aumentando però notevolmente la complessità computazionale degli algoritmi. Quello che si fa è scegliere lo stato  $x_t$  in modo tale che l'effetto di variabili non modellate sia trascurabile.

Adottando tale ipotesi possiamo scrivere la seguente uguaglianza

$$(2.2) \quad p(x_t | x_{0:t-1}, z_{1:t-1}, u_{1:t}) = p(x_t | x_{t-1}, u_t)$$

ovvero la probabilità a priori è statisticamente indipendente dai dati ottenuti nell'intervallo  $(0, t-1]$ . Tale funzione prende il nome di *probabilità di transizione dello stato* o semplicemente *modello del moto* e indica la probabilità che un controllo

$u_t$  induca una transizione dallo stato  $x_{t-1}$  allo stato  $x_t$ . La stessa ipotesi può essere applicata ad un altro modello ottenendo

$$p(z_t | x_{0:t}, z_{1:t-1}, u_{1:t}) = p(z_t | x_t)$$

Tale densità è detta *modello generativo della misura* o *modello del sensore* e indica come le misure siano generate dalla conoscenza dello spazio operativo. In questo senso le misure sono viste come proiezioni rumorose dell'ambiente. L'equazione 2.1 diventa

$$bel(x_t) = \eta p(z_t | x_t) p(x_t | z_{1:t-1}, u_{1:t})$$

Il termine  $p(x_t | z_{1:t-1}, u_{1:t})$  viene indicato con il simbolo  $\overline{bel}(x_t)$  e rappresenta la stima dello stato prima di incorporare la misura al tempo  $t$ . Applichiamo il *teorema delle probabilità totali* ottenendo:

$$\begin{aligned} \overline{bel}(x_t) &= p(x_t | z_{1:t-1}, u_{1:t}) = \int_{\Omega} p(x_t | x_{0:t-1}, z_{1:t-1}, u_{1:t}) p(x_{t-1} | z_{1:t-1}, u_{1:t}) dx_{t-1} \\ &= \int_{\Omega} p(x_t | x_{t-1}, u_t) p(x_{t-1} | z_{1:t-1}, u_{1:t}) dx_{t-1} \\ &= \int_{\Omega} p(x_t | x_{t-1}, u_t) bel(x_{t-1}) dx_{t-1} \end{aligned}$$

considerando la (2.2), ricordando che  $p(A, B, C) = p(C)p(B|C)p(A|B, C)$  e omettendo  $u_t$ .

Abbiamo ottenuto così la regola di aggiornamento del filtro di Bayes:

**PREDIZIONE**  $\overline{bel}(x_t) = \int_{\Omega} p(x_t | x_{t-1}, u_t) bel(x_{t-1}) dx_{t-1}$

**CORREZIONE**  $bel(x_t) = \eta p(z_t | x_t) \overline{bel}(x_t)$

Il primo step è il *passo di predizione* e calcola la stima dello stato al tempo  $t$  partendo dalla stima dello stato al tempo  $t-1$  considerando solo il modello del moto, senza incorporare la misura.  $\Omega$  rappresenta lo spazio di stato del robot, ovvero lo spazio di tutti i possibili valori assunti dallo stato del robot. Il secondo step è il passo di correzione o aggiornamento e la stima ottenuta precedentemente viene affinata integrando la misura  $z_t$ . Notiamo che le due relazioni necessitano di tre elementi per agire nel modo corretto: una stima iniziale  $bel(x_0)$ , il modello del moto e il modello della misura. La stima iniziale viene modellata con una funzione densità discreta centrata su  $x_0$  se nota, altrimenti se sconosciuta assume una distribuzione uniforme su tutto lo spazio degli stati (livello di incertezza massimo).

Lo stato di un sistema robotico mobile è costituito dalla posa (posizione e orientamento) del robot nel sistema di coordinate scelto e dalla mappa dell'ambiente. Pertanto l'equazione di stima può essere riscritta nel modo seguente indicando con  $p_t$  la posa e con  $m_t$  la mappa al tempo  $t$  :

$$bel(p_t, m_t) = \eta \cdot p(z_t | p_t, m_t) \cdot \iint p(p_t, m_t | p_{t-1}, m_{t-1}, u_t) bel(p_{t-1}, m_{t-1}) dp_{t-1} dm_{t-1}$$

Adottando l'ipotesi di ambiente statico e considerando il moto del robot indipendente dalla mappa, possiamo ridurre la dimensione del problema :

$$bel(p_t, m) = \eta \cdot p(z_t | p_t, m) \cdot \int p(p_t | p_{t-1}, u_t) bel(p_{t-1}, m) dp_{t-1}$$

Tutti i modelli presentati in questa tesi faranno l'assunzione che l'ambiente sia statico. La generazione di modelli dinamici prevede l'uso di altre tecniche (ad esempio reti neurali) che qui non tratteremo. Descriviamo ora brevemente i rimanenti due modelli stocastici.

### 2.3.2 Modello del moto

Il modello del moto è rappresentato dalla funzione densità di probabilità  $p(x_t | x_{t-1}, u_t)$  e indica la probabilità che un controllo  $u_t$  possa produrre un passaggio di stato da  $x_{t-1}$  a  $x_t$ . Tali modelli sono ricavati aggiungendo rumore ai modelli ideali della cinematica. Il rumore viene modellato come una distribuzione di probabilità a media nulla e varianza adeguata. In base al tipo di grandezza fisica usata come controllo abbiamo diversi modelli. A titolo di esempio presentiamo un modello del moto basato sull'odometria.

Gli odometri sono sensori propriocettivi, ovvero misurano lo stato interno del robot (posizione, velocità, ecc) integrando i dati ottenuti dalla misura delle rotazioni delle ruote. Non sono segnali di controllo ma misurazioni, ma vengono usati come tali per ridurre lo spazio di stato. Consideriamo un robot che si muova in un ambiente bidimensionale supposto statico. Indichiamo con  $x$  la posa del robot in un generico istante. Nel piano un robot ha tre gradi di libertà pertanto la sua posa sarà descritta da due coordinate cartesiane e un angolo che ne individua l'orientamento rispetto all'asse positivo delle ascisse (*bearing*). Supponiamo che il robot si sia spostato da  $x_{t-1}$  a  $x_t$  nel sistema di riferimento fisso a cui corrispondono le coordinate  $\bar{x}_{t-1}$  e  $\bar{x}_t$  ottenute dai segnali odometrici. Ovviamente la relazione tra i due sistemi di riferimento è sconosciuta, ma possiamo considerare la differenza tra  $\bar{x}_{t-1}$  e  $\bar{x}_t$  come una buona approssimazione della differenza tra  $x_{t-1}$  e  $x_t$ . Il segnale di controllo corrispondente sarà dun-

que  $u_i = (\bar{x}_{i-1}, \bar{x}_i)$ . Per estrarre l'informazione, lo spostamento viene diviso in una rotazione  $\delta_1$ , una traslazione  $\delta_2$  ed una rotazione  $\delta_3$ . Si può dimostrare ricordando il Teorema di Chasles ("Ogni spostamento rigido è una rotazione attorno ad un asse e una traslazione lungo lo stesso asse") che ad ogni coppia  $(\bar{x}_{i-1}, \bar{x}_i)$  corrisponde un'unica terna di parametri  $(\delta_1, \delta_2, \delta_3)$ . Una delle possibili realizzazioni di tale modello è quella di utilizzare il *modello inverso del moto*: un algoritmo che riceve in ingresso le due posizioni assolute  $(x_{i-1}, x_i)$  e odometriche  $(\bar{x}_{i-1}, \bar{x}_i)$  restituendo la probabilità di errore. Assumendo indipendenti le cause di rumore e definiti alcuni parametri che indicano l'accuratezza del robot, la densità di probabilità è calcolata come il prodotto delle tre densità riferite ad ogni  $\delta_i$ .

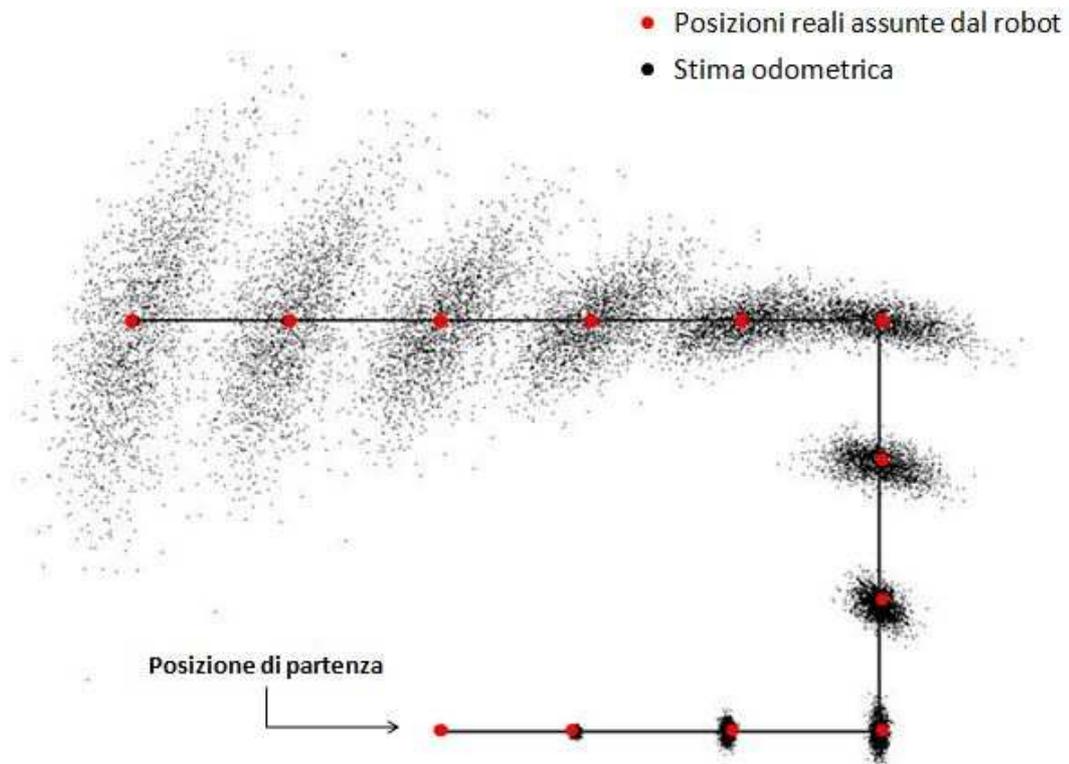


Figura 2.4 Stima della posizione mediante odometria [1]

La figura 2.4 mostra la stima della posizione del robot sfruttando solo l'odometria. Si noti che l'incertezza aumenta con l'aumentare della distanza percorsa. Ciò indica che la sola odometria non è sufficiente a stimare lo stato di un robot, ma occorre integrare un opportuno modello di misura che effettui un'adeguata correzione.

### 2.3.3 Modello del Sensore

Nella regola di aggiornamento del filtro di Bayes un altro importante concetto è il modello del sensore  $p(z_t | x_t)$  che descrive il processo mediante il quale sono generate le misure dei sensori esteroceettivi, ovvero che rilevano i cambiamenti esterni al robot (laser, sonar, ecc). Esso indica la probabilità che un sensore ha di effettuare una misura  $z_t$  sulla base di tutti gli stati possibili. Per capire meglio come ciò avviene facciamo un piccolo esempio:  $x_t$  rappresenta lo stato di una porta e di conseguenza  $\Omega = \{ \text{"aperta"}, \text{"chiusa"} \}$ . Supponiamo di aver rilevato che

$$\begin{aligned} p(z_t = \text{"aperta"} | x_t = \text{"aperta"}) &= 0.8 \\ p(z_t = \text{"aperta"} | x_t = \text{"chiusa"}) &= 0.4 \end{aligned}$$

in questo caso il modello del sensore è una funzione densità discreta che ci indica che il sensore rileva la porta "aperta" quando essa è "aperta" l'80% delle volte (quindi la percentuale di errore nel rilevare una porta aperta è del 20%) e rileva una porta "chiusa" quando essa è "chiusa" il 60% delle volte (quindi la percentuale di errore nel rilevare una porta chiusa è del 40%)

Notare che  $x_t$  rappresenta lo stato del sistema (posa e mappa) e non solo la posa. Per evitare confusione indicheremo tale funzione densità con  $p(z_t | x_t, m)$  dove  $x_t$  è la posa e  $m$  la mappa statica dell'ambiente.

La determinazione del modello dipende ovviamente dal tipo di sensore

che utilizziamo. Ne distinguiamo due grandi famiglie:

- rilevatori di distanza (sonar, laser), il sensore rileva la distanza dagli ostacoli presenti nell'ambiente;
- misura di landmark, il sensore misura la distanza da punti di riferimento (landmark) che caratterizzano l'ambiente la cui posizione è nota.

Nelle simulazioni effettuate per questa tesi abbiamo ipotizzato di servirci di un robot munito di rilevatori di distanza. Mostriamo ora come ricavare un modello adatto a questo tipo di sensori. In genere un robot non è munito di un singolo sensore. Per integrare tutte le misure fatte dai diversi sensori si ipotizza che tali misure siano indipendenti l'una dall'altra e la densità di probabilità totale è calcolata come produttoria delle singole densità. Tra i fattori che producono la violazione di tale ipotesi citiamo: la presenza di persone nelle vicinanze del sensore, errori nel modellare la mappa, approssimazioni nelle funzioni densità di probabilità.

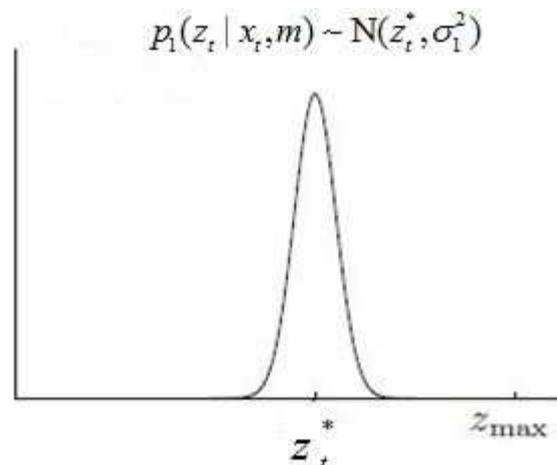


Figura 2.5 Distribuzione Normale [1]

Un modo per rappresentare la densità  $p(z_t | x_t, m)$  è quello di considerare le condizioni in cui un sensore deve operare e modellare separatamente gli elementi che concorrono alla creazione della misura per poi combinarli linearmente. Un primo aspetto da

modellare è il rilevamento dalla misura: ogni sensore è in grado di ottenere una distanza che si discosta sensibilmente da quella vera a causa della intrinseca imprecisione dello strumento. Tale caratteristica è modellata con una Gaussiana centrata sul valore vero  $z_t^*$  della distanza e con varianza  $\sigma_1^2$  come mostrato nella figura 2.5.

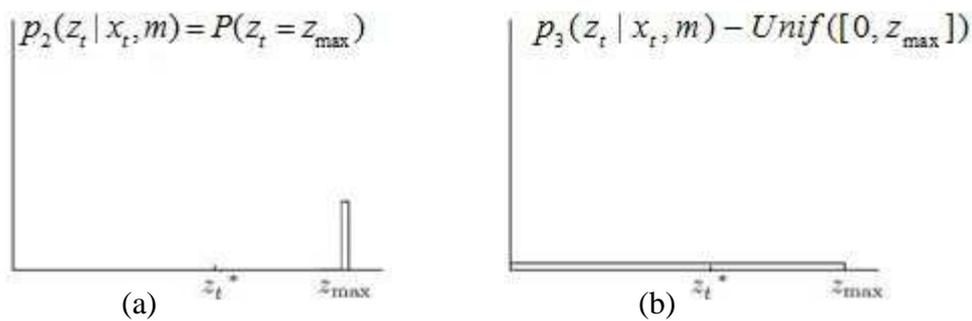


Figura 2.6 Distribuzione discreta e uniforme [1]

Un secondo aspetto è l'impossibilità di un sensore di portare a termine una misura in modo corretto. Quando ciò accade lo strumento restituisce come valore letto il suo range massimo. Tale caratteristica è modellata con una distribuzione discreta che assume valore unitario in corrispondenza di  $z_{\max}$  (vedi fig.2.6a). Un terzo aspetto sono le misure casuali: i rilevatori di distanza producono durante il loro funzionamento delle misure inaspettate.

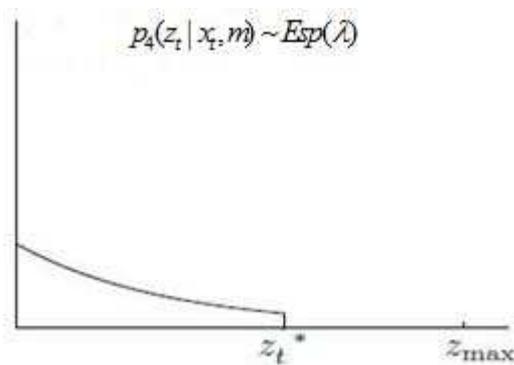


Figura 2.7 Distribuzione esponenziale [1]

Ciò è modellato con una distribuzione uniforme sull'intervallo  $[0, z_{\max}]$  di valore  $1/z_{\max}$  (vedi fig.2.6b).

L'ultimo elemento riguarda gli oggetti inaspettati che si trovano nel range di rilevamento del sensore. Poiché il nostro modello ipotizza l'ambiente statico tutto ciò che si muove, come le persone, viene considerato rumore aggiunto alla misura. Tale aspetto viene modellato con una distribuzione esponenziale di parametro  $\lambda$  (vedi fig. 2.7). Il modello del sensore  $p(z_t | x_t, m)$  si calcola come una combinazione lineare delle precedenti quattro distribuzioni e un possibile risultato è mostrato nella fig. 2.8.

$$p(z_t | x_t, m) = \alpha_1 p_1(z_t | x_t, m) + \alpha_2 p_2(z_t | x_t, m) + \alpha_3 p_3(z_t | x_t, m) + \alpha_4 p_4(z_t | x_t, m)$$

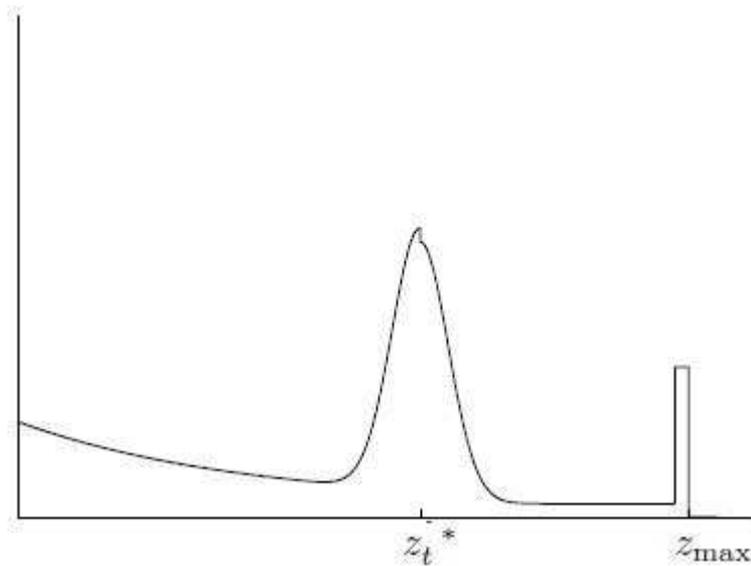


Figura 2.8 Modello del sensore [1]

Affinché il modello calcolato sia una funzione densità di probabilità la somma dei coefficienti  $\alpha_i$  deve essere uno. Il valore di tali coefficienti così come i parametri delle diverse distribuzioni vengono calcolati attraverso prove empiriche oppure grazie all'ausilio di algoritmi che tra tutti i possibili valori scelgono quelli che massimizzano la densità  $p(z_t | x_t, m)$  (Nella notazione compatta della formula di Bayes, tale densità prende il nome di “Likelihood”; tali stimatori prendono dunque il nome di “Maximum

Likelihood Estimator” o più semplicemente ML estimator). In molte applicazioni pratiche per evitare di calcolare il prodotto di quattro densità, il modello del sensore assume una distribuzione normale a media nulla e varianza  $\sigma^2$ . Tuttavia più un sensore è modellato in maniera accurata, più i dati calcolati saranno precisi.

## 2.4 Filtri di Bayes Approssimati

Nel precedente capitolo abbiamo indicato il filtro di Bayes come un metodo per poter risolvere il problema della stima stocastica. Osservando attentamente la regola di aggiornamento di tale filtro ci accorgiamo che essa non può essere applicata sempre, ma solo quando l’integrale e la moltiplicazione possono essere calcolati in forma chiusa oppure quando  $\Omega$  ha una dimensione finita. Negli altri casi dobbiamo necessariamente introdurre delle ipotesi aggiuntive e in base a tali ipotesi distinguiamo vari approcci.

Esistono due grandi famiglie:

- Filtri Gaussiani
- Filtri non parametrici

I filtri gaussiani prendono il loro nome dal fatto che tutti i modelli presenti nel filtro di Bayes, la distribuzione della stima iniziale e il rumore assumono una distribuzione Gaussiana. I filtri gaussiani applicati ad un sistema lineare e con uno spazio di stato  $\Omega$  continuo vengono chiamati *Filtri di Kalman (KF)*, che lavorano aggiornando progressivamente la media e la covarianza della gaussiana di partenza. Il filtro di Kalman è uno degli strumenti più importanti nel calcolo della stima stocastica in quanto è possibile dimostrare che esso minimizza l’errore quadratico medio e massimizza la densità di probabilità a posteriori (la stima). Il filtro di Kalman può essere esteso al caso non lineare e in base al metodo usato per linearizzare le non-linearità distinguiamo l’Extended Kalman Filter (EKF) e l’Unscented Kalman Filter(UKF). EKF usa come

metodo di linearizzazione l'espansione in serie di Taylor e linearizza fino al primo ordine. UKF usa come metodo di linearizzazione la trasformata "Unscented" che equivale ad una linearizzazione fino al secondo ordine dell'espansione di Taylor. Uno dei vantaggi maggiori del filtro di Kalman è la velocità di elaborazione. Lo svantaggio principale di questo metodo sta nell'assunzione di densità unimodali che mal rappresentano la realtà. Considerando un'osservazione di due punti indistinguibili, la densità di probabilità risultante sarà una funzione multimodale (nel caso specifico bimodale) che viola l'ipotesi di funzioni gaussiane del filtro. Tali filtri sono adatti a risolvere problemi come la localizzazione locale, ma inadatti per problemi quali localizzazione globale e data association problem. Tale inconveniente è superato con l'uso dei Filtri non parametrici che non usano una forma standard di funzione densità, ma la approssimano impiegando un numero finito di parametri. Appartengono alla classe dei filtri non parametrici i Filtri Particellari (Particle Filter) e i Filtri a Istogramma. I primi approssimano il modello con un numero casuale di campioni che si addensano dove la probabilità è maggiore. I secondi dividono lo spazio di stato in un numero finito di intervalli ed assegnano ad ogni intervallo una probabilità. Queste tecniche hanno un forte potere di rappresentazione della realtà e sono usate quando i modelli forniscono ipotesi distinte e separate. Il grande svantaggio risiede nella complessità computazionale, basti pensare che un filtro particellare è diversi ordini di grandezza più pesante di un filtro di Kalman.

Nel problema della mappatura con griglie di occupazione viene usato un particolare filtro a istogramma, il filtro binario di bayes. Presentiamo ora una breve descrizione del suo funzionamento.

## 2.4.1 Filtro Binario di Bayes

Il filtro a Istogramma divide lo spazio di stato in un numero finito di regioni e assegna ad ognuna una probabilità. Esso rappresenta un metodo di approssimazione di uno spazio continuo (vedi fig. 2.9). Quando è applicato ad uno spazio discreto viene indicato con il termine *Filtro Discreto di Bayes*.

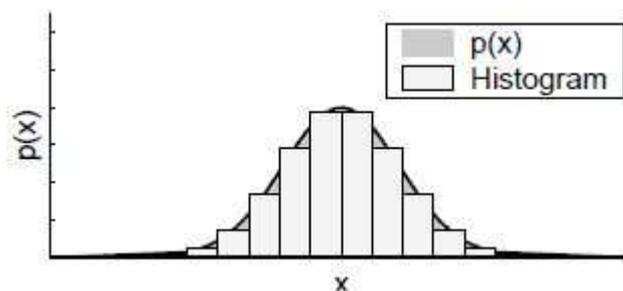


Figura 2.9 Approssimazione di una Gaussiana [1]

Grazie all'utilizzo di questo metodo lo spazio di stato  $\Omega$  continuo e infinito si trasforma in uno spazio discreto e finito e l'integrale nella regola di aggiornamento viene sostituito da una sommatoria. Nel problema della mappatura con griglie di occupazione, l'ambiente viene diviso in un numero finito di celle e ogni cella (lo stato) può assumere solo due valori (1 se occupata, 0 se libera). Tali problemi vengono affrontati con l'uso del *Filtro di Bayes Binario*. Mostriamo come ottenere un'implementazione elegante di tale filtro. Se assumiamo che lo stato del sistema sia statico, la stima dipenderà solo dalle misure e non dal tempo

$$bel(x) = p(x | z_{1:t}, u_{1:t}) = p(x | z_{1:t})$$

tale modello viene detto *modello inverso del sensore*. Se lo stato è statico, assumerà i valori  $x$  e  $x^c$  ( $x$  complementare). In queste condizioni il filtro viene implementato nella forma *log odds ratio* con

$$l_t(x) = \log \left( \frac{p(x | z_{1:t})}{1 - p(x | z_{1:t})} \right)$$

e assume la seguente forma

$$l_t(x) = l_{t-1}(x) + \log \left( \frac{p(x | z_t)}{1 - p(x | z_t)} \right) - l_0(x)$$

Le proprietà di questo algoritmo sono:

- È additivo, in quanto calcola la stima come somma di altri tre termini;
- Usa il modello inverso del sensore, usato quando le misure sono più complicate dello stato;
- $l_0(x) = \log \left( \frac{p(x)}{1 - p(x)} \right)$  con  $p(x)$  probabilità a priori.



# Capitolo 3

## Tecniche di Mappatura

### 3.1 Le griglie occupazionali

Lo scopo di questo capitolo è fornire un esempio di come sia possibile ricavare una mappa dell'ambiente partendo da un insieme di misure incerte e rumorose. In molte applicazioni robotiche la conoscenza della mappa non è disponibile a priori e dotare un robot mobile di un metodo per acquisirla consente di adattarsi ai cambiamenti senza l'intervento umano. Acquisire una mappa non è un compito semplice: le mappe sono definite su uno spazio continuo quindi l'insieme di tutte le mappe ha dimensione infinita. La complessità di tale problema dipende da numerosi fattori:

- Dimensione dello spazio da mappare;
- Presenza di rumore nei sensori;
- Data Association Problem (stabilire che i dati si riferiscono allo stesso oggetto osservato in due istanti diversi di tempo(chiusura del cerchio))

La costruzione della mappa (Mapping) non può essere separato dalla conoscenza della posizione del robot all'interno di tale mappa (Localizzazione). Per poter trattare il mapping come un problema isolato dobbiamo assumere una prima ipotesi di partenza:

**I** ipotesi **➔** Le posizioni assunte dal robot sono note

Tale assunzione potrebbe sembrare troppo forte, ma ci permette di arrivare a soluzioni molto importanti nelle applicazioni pratiche soprattutto quelle orientate al path planning e alla navigazione. Tutti gli algoritmi di mappatura vengono utilizzati nella fase di post-elaborazione, questo perché gli algoritmi di SLAM non sono in grado di generare mappe adatte alla navigazione. La figura 3.1 mostra una mappa ottenuta con un algoritmo di

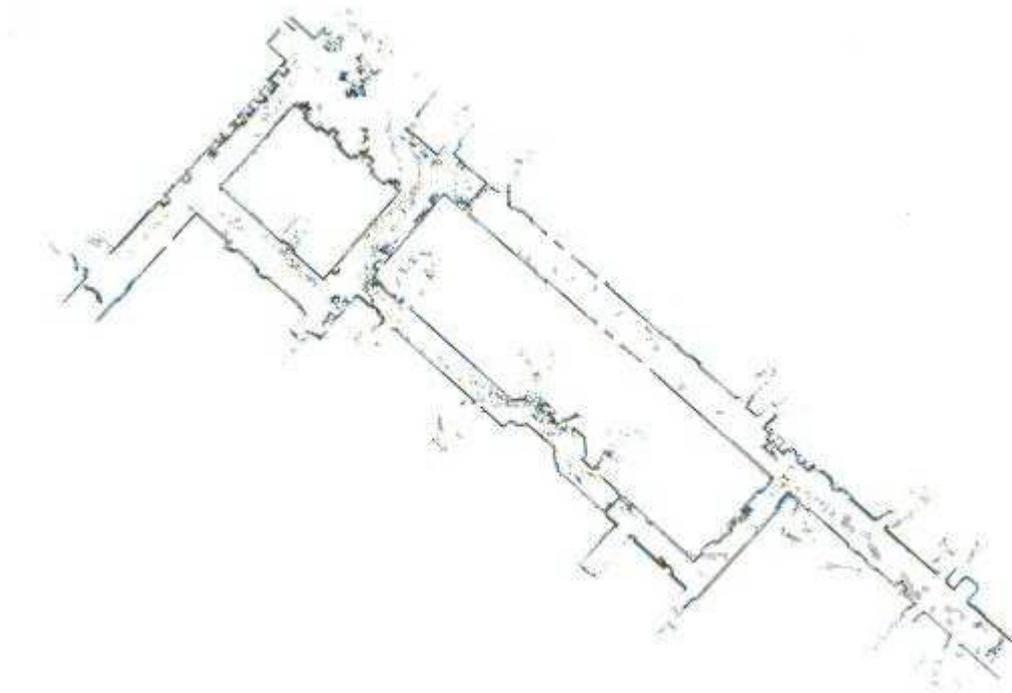


Figura 3.1 Mappa costruita attraverso algoritmi di SLAM [1]

SLAM. La maggior parte dei dati è danneggiata dalla presenza delle persone (puntini in grigio nella figura) o altri oggetti in movimento. Un algoritmo di path planning avrebbe difficoltà a trovare un percorso attraverso tutti questi ostacoli. La figura 3.2 mostra invece il risultato di un algoritmo di mappatura partendo da un insieme di dati grezzi forniti dallo SLAM. L'azione filtrante svolta dall'algoritmo rende la mappa adatta alla navigazione.

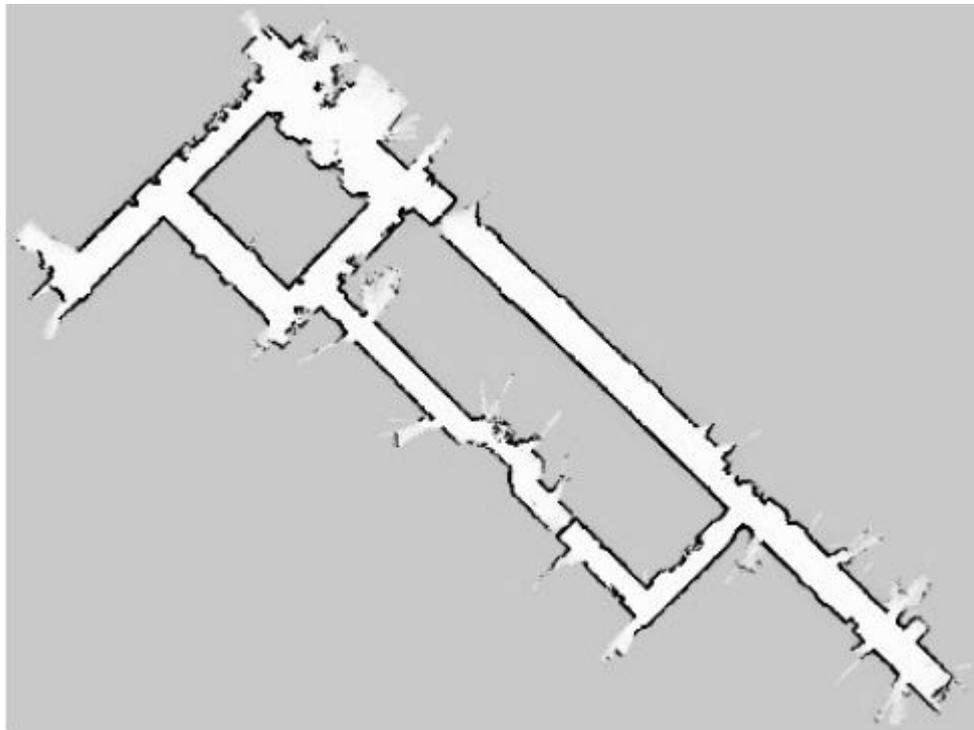
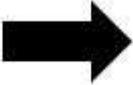


Figura 3.2 Mappa costruita attraverso algoritmi di Mapping [1]

Un approccio molto popolare usato per risolvere il problema del mapping è l'impiego della *mappa basata su griglie occupazionali (Occupancy Grid Map)*. Tale metodo prevede la suddivisione dello spazio operativo in un numero finito di celle. Ad ogni cella viene associata una variabile binaria che ne indica lo stato di occupazione

(1=occupata, 0=libera). Ricordando che nel problema del mapping lo stato incognito è rappresentato dalla sola mappa al tempo  $t$  ( $m_t$ ), avendo ipotizzato note le posizioni assunte, esso equivale a stimare la probabilità che una posizione sia occupata da un ostacolo, ovvero stimare la funzione densità  $p(m_t | z_{1:t}, x_{1:t}, u_{1:t})$ , dove  $x_{1:t}$  è il cammino del robot definito come sequenza delle sue pose. Per ridurre il grado di difficoltà del problema facciamo un'altra ipotesi che si verifica spesso in molte applicazioni pratiche:

**II** **Ipotesi**  Mappa statica

Grazie all'assunzione di stato (= mappa) statico la stima risulta indipendente dai segnali di controllo  $u_{1:t}$  e dal tempo, valendo la relazione

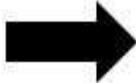
$$p(m_t | z_{1:t}, x_{1:t}, u_{1:t}) = p(m | z_{1:t}, x_{1:t})$$

Con la dicitura “mappa statica” intendiamo che la configurazione dell'ambiente (muri, porte, ecc) non cambia nel tempo e tutti gli oggetti in movimento saranno trattati come rumore aggiunto alle misure.

Il calcolo della densità  $p(m | z_{1:t}, x_{1:t})$  può avvenire in diversi modi. Nei prossimi paragrafi ne presenteremo due: il primo introduce una terza ipotesi restrittiva e fa uso del filtro binario di Bayes; il secondo utilizza una tecnica di massimizzazione per trovare la mappa più probabile.

## 3.2 Mappatura basata su Filtro Binario

La risoluzione del problema del mapping mediante questo approccio prevede

**III** **Ipotesi**  Indipendenza delle celle

l'introduzione di una terza ipotesi:

Tale assunzione ci permette di dividere il problema in tanti piccoli sottoproblemi: ottenere una mappa si trasforma nel problema di stimare la probabilità di occupazione di ogni singola cella.

Sia  $m_i$  la  $i$ -esima cella tale che  $(m_1, m_2, \dots, m_N) = m$  con  $N$  numero totale delle celle. Assunta valida la precedente ipotesi vale la relazione:

$$p(m | z_{1:t}, x_{1:t}) = \prod_{i=1}^N p(m_i | z_{1:t}, x_{1:t})$$

Sotto queste tre ipotesi è possibile utilizzare il filtro binario. Nella forma log odds ratio:

$$l_t(m_i) = l_{t-1}(m_i) + \log \left( \frac{p(m_i | z_t, x_t)}{1 - p(m_i | z_t, x_t)} \right) - l_0(m_i)$$

La quantità  $\log \left( \frac{p(m_i | z_t, x_t)}{1 - p(m_i | z_t, x_t)} \right)$  prende il nome di *modello inverso del sensore nella*

*forma log odds ratio* ed è la parte più critica di tutto l'algoritmo in quanto si occupa di valutare lo stato di occupazione di ogni cella. Per fare ciò usa tre valori  $l_{free}, l_0, l_{occ}$  con  $l_{free} < l_0 < l_{occ}$  (ad esempio  $l_{free} = -10, l_0 = 0, l_{occ} = 10$ ) che indicano rispettivamente "cella libera", "non interessata dal cono del sensore", "cella occupata". Nel primo pas-

so l'algoritmo individua la distanza del centro del robot dal centro di massa della cella  $i$ -esima. Nel secondo passo calcola il sensore più vicino alla cella. Questo step viene eseguito considerando una serie di angoli (vedi fig. 3.3):

- $\theta$  è l'angolo che l'asse delle ascisse forma con la direzione principale del robot;
- $\gamma$  è l'angolo che l'asse delle ascisse forma con la retta che unisce il centro del robot e il centro della cella  $i$ -esima;
- $\phi = \gamma - \theta$  è l'angolo che la direzione principale del robot forma con la retta che unisce il centro del robot e il centro della cella  $i$ -esima;
- $\theta_{k,sens}$  è l'angolo che la direzione principale del robot forma con il sensore  $k$ -esimo;
- $\varepsilon = \phi - \theta_{k,sens}$  è l'angolo che la retta che unisce i centri del robot e della cella forma con il fascio del sensore.

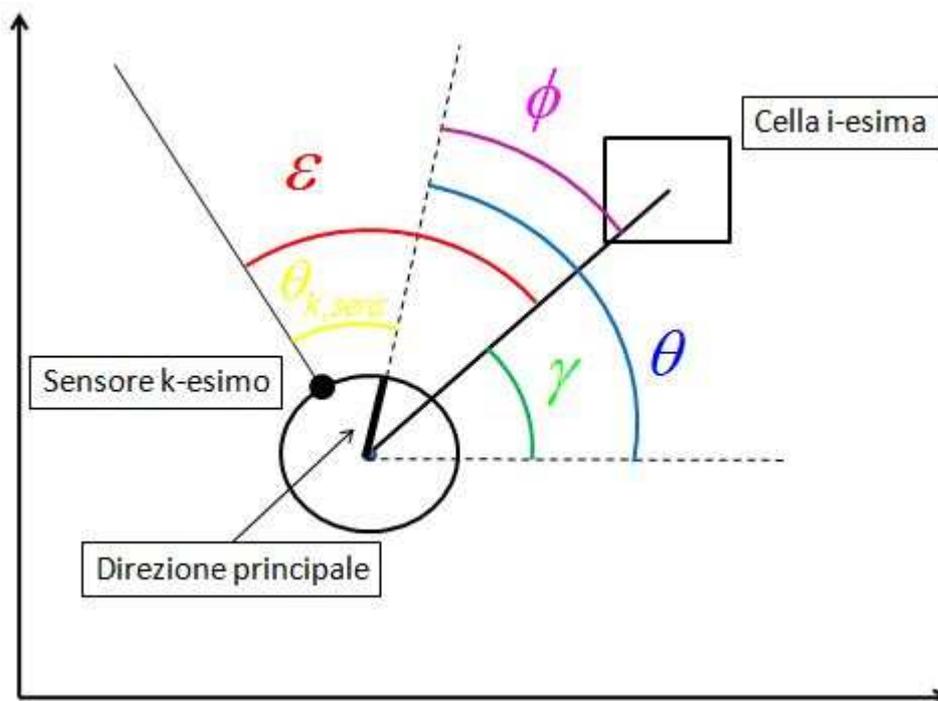


Figura 3.3 Angoli caratteristici

Per ogni cella il sensore più vicino sarà quello con  $\epsilon$  minore.

L'ultimo passo è valutare lo stato di occupazione delle celle: se la cella non si trova nel cono del sensore assume valore  $l_0$ ; se la cella si trova nel cono del sensore, la misura è minore del range massimo del sensore e l'ostacolo è per il 50% all'interno della cella, assume valore  $l_{occ}$ ; altrimenti assume valore  $l_{free}$ .

Per capire meglio come lavora quest'algoritmo facciamo un esempio. Consideriamo la situazione mostrata in fig. 3.4.

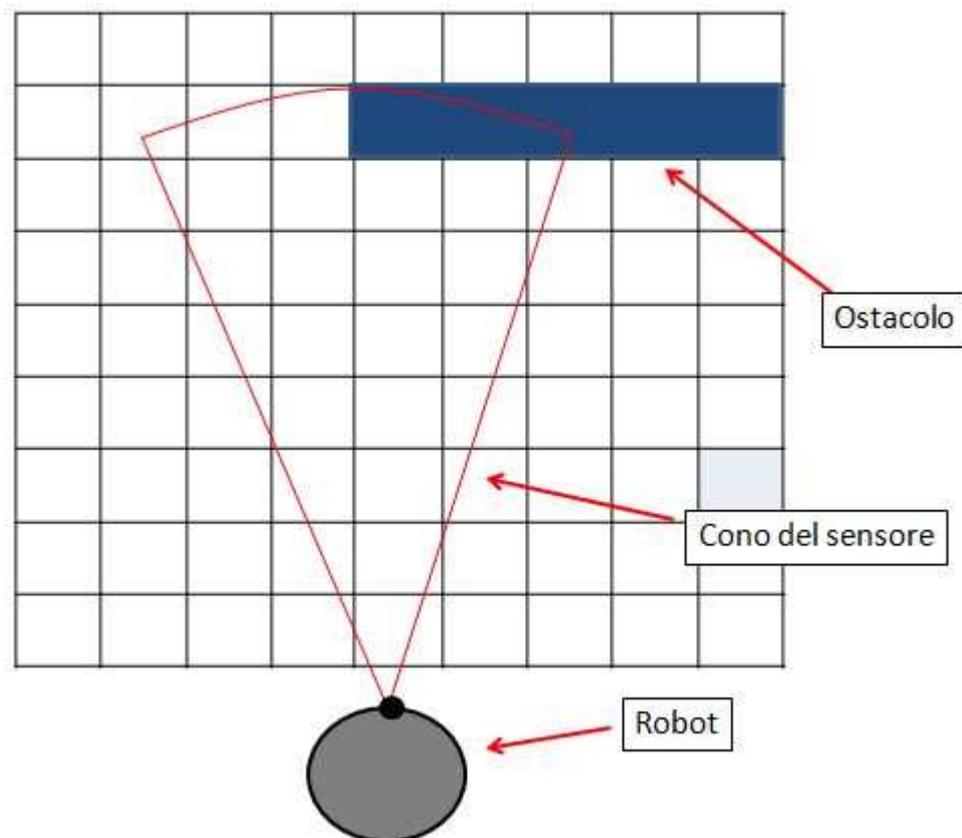


Figura 3.4 Ambiente



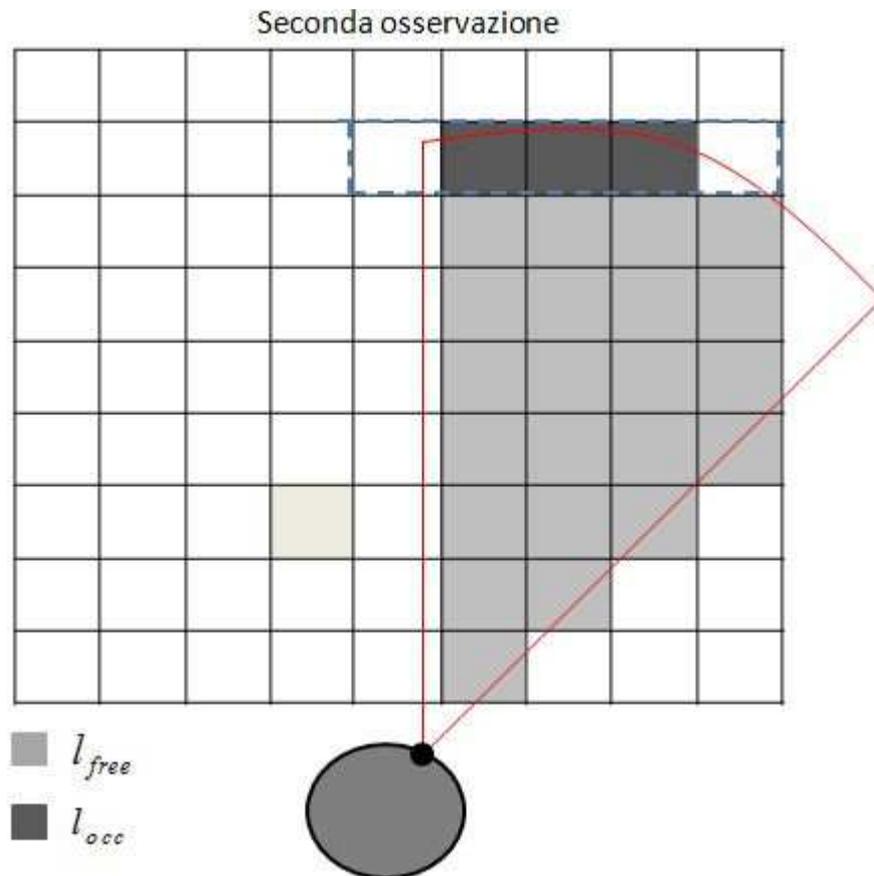


Figura 3.6 Seconda misura del sensore. La cella è considerata interessata dal cono del sensore se la investe almeno per il 50% (tali valori possono essere modificati settando alcuni parametri)

Al termine delle misure ogni cella avrà un valore di occupazione ricavato come somma dei valori ottenuti ad ogni misurazione distinta. Come mostra la figura 3.7, questo aspetto fa sì che celle osservate più frequentemente tendano ad assumere valori vicino agli estremi (1 o 0). Ciò si verifica facilmente ricordando che la stima è legata alla forma log odds ratio dalla relazione:

$$bel(x) = 1 - \frac{1}{1 - e^t}$$

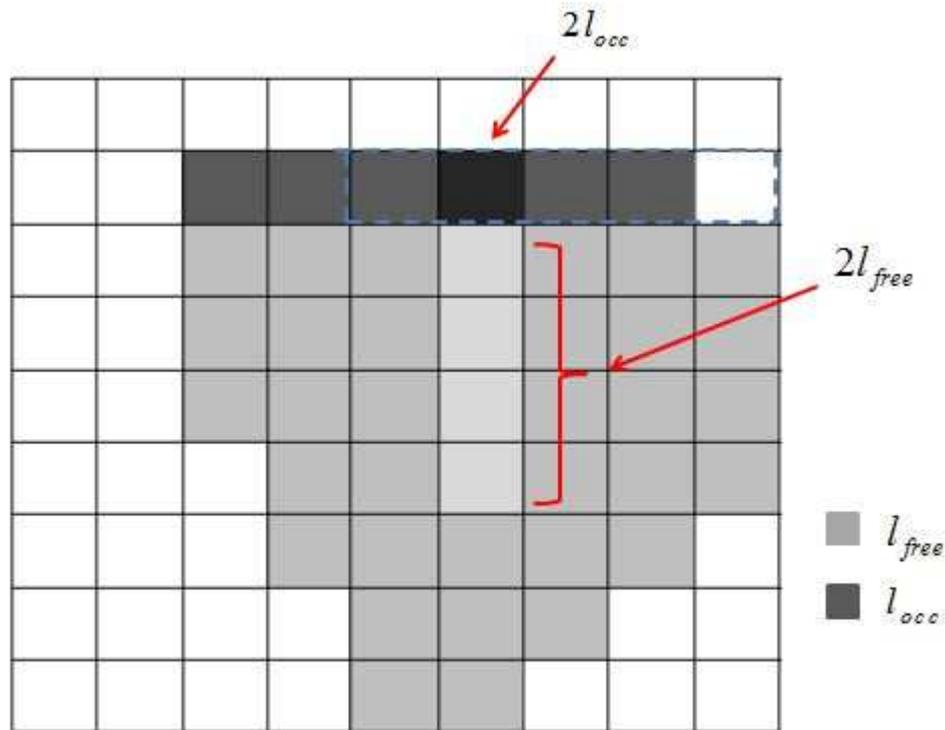


Figura 3.7 Mappa ottenuta dopo le due misurazioni

Abbiamo mostrato i due difetti principali di questo algoritmo: il primo evidenzia che se un ostacolo occupa solo una parte dell'arco del cono del sensore, tutte le celle dell'arco vengono etichettate come occupate; il secondo è che tale situazione è gestita grossolanamente sommando i valori assegnati dalla cella nelle misure precedenti dando vita così a risultati fasulli (una cella potrebbe essere considerata occupata quando in realtà non lo è. Questo accade nella fig. 3.5: le due celle in alto a sinistra vengono etichettate con  $l_{occ}$  quando in realtà sono libere). In altre parole il fatto che il cono del sensore spazi su diverse celle induce importanti dipendenze tra le suddette celle. Quando decomponiamo il problema nell'unione di tanti sotto-problemi indipendenti, perdiamo la possibilità di rappresentare queste dipendenze. Un modo per risolvere questo aspetto è fare in modo che l'arco ricopra solo un piccolo numero di celle. Una soluzione più efficiente è invece quella che si basa sull'utilizzo degli stimatori Maximum a Posteriori presentati nel prossimo paragrafo.

### 3.3 Mappatura basata su stimatore MAP

Come abbiamo già precisato nel paragrafo 2.3, uno stimatore stocastico è un metodo per calcolare la stima di una quantità basandosi sui dati di un'osservazione affetta da rumore. Lo stimatore è pertanto una variabile casuale espressa in funzione dei dati e una sua realizzazione è detta stima ( $\hat{\theta}(x)$ ). Esistono diversi modi per costruire un osservatore e tutti i differenti stimatori possono essere giudicati confrontando alcune proprietà. Le tre più importanti classi di stimatori sono: lo stimatore LSQ, lo stimatore ML e lo stimatore MAP. Appartengono alla classe degli stimatori LSQ (Least Square) tutti quelli che trovano la stima che minimizza l'errore quadratico medio:

$$\hat{\theta}_{LSQ}(x) = \arg \min_{\hat{\theta}(x)} E[(\hat{\theta}(x) - \theta)^2]$$

L'errore quadratico medio indica in media quanto lontana è la mia stima dal parametro da stimare. Uno stimatore è detto invece ML (Maximum Likelihood) se massimizza la probabilità di ottenere una misura conoscendo gli stati del sistema:

$$\hat{\theta}_{ML}(x) = \arg \max_x p(z | x)$$

Infine uno stimatore è detto MAP (Maximum A Posteriori) se massimizza la probabilità di ottenere una stima partendo dalla conoscenza delle misure:

$$\hat{\theta}_{MAP}(x) = \arg \max_x p(x | z)$$

Nel precedente paragrafo abbiamo risolto il problema del mapping, che equivale al calcolo della densità  $p(m | z_{1:t}, x_{1:t})$ , dividendolo in un insieme di elementi indipendenti. Tale approccio portava alla realizzazione di mappe la cui accuratezza dipendeva molto dal numero di osservazioni effettuate. Infatti tutti i difetti del precedente algoritmo si riducono notevolmente quando il numero delle osservazioni cresce molto. In parole semplici se il nostro robot esplora tutto l'ambiente, anche il più piccolo anfratto, per un numero considerevole di volte, la mappa tende ad essere abbastanza accura-

ta. Nella realtà questa situazione è molto improbabile: anche se esistono parti dell'ambiente in cui un robot potrebbe fermarsi a lungo (spazi larghi), la maggior parte dei luoghi sono solo attraversati dal robot o per nulla esplorati (corridoi e piccole rientranze).

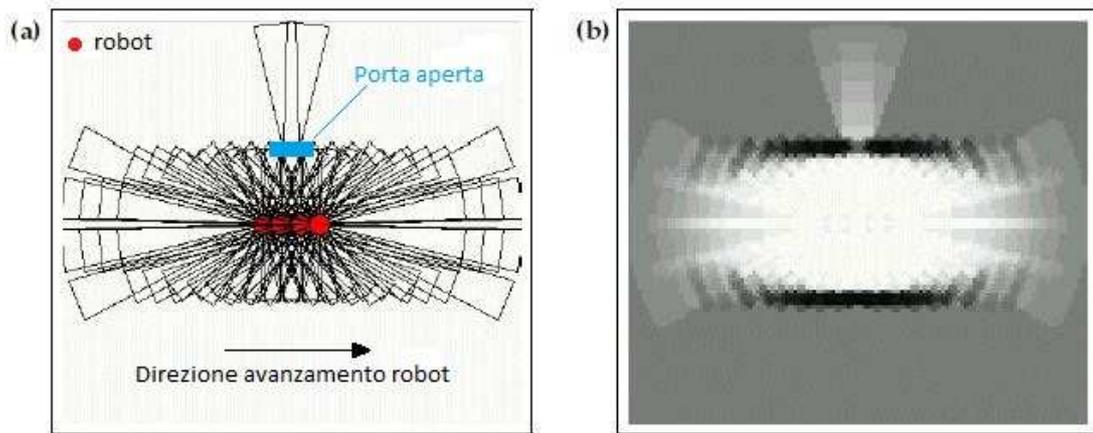


Figura 3.8 La mappa non riesce a rappresentare la porta [1]

La figura 3.8 mostra un esempio di questa limitazione: un robot avanza lungo un corridoio dove è presente una porta aperta (fig. 3.8a); la mappa risultante ottenuta mediante il primo approccio non riesce a rappresentare lo stato della porta (fig. 3.8b). Ulteriori esempi relativi a questa limitazione sono forniti nella parte dedicata alle simulazioni.

Un secondo metodo che realizza mappe più accurate si fonda sulla realizzazione di uno stimatore MAP basato sul modello della misura. Infatti poiché il numero di mappe che può essere definito su una griglia è troppo grande,  $p(m | z_{1:t}, x_{1:t})$  non può essere calcolata analiticamente. Quello che si fa è *cercare tra tutte le possibili mappe quella ottima  $m^*$  che massimizza la probabilità a posteriori  $p(m | z_{1:t}, x_{1:t})$*  ovvero:

$$m^* = \arg \max_m p(m | z_{1:t}, x_{1:t})$$

Questo metodo non restituisce la forma della densità  $p(m | z_{1:t}, x_{1:t})$ , ma una delle sue mode (i.e. un massimo locale).

### 3.3.1 Costruzione dello Stimatore MAP

Consideriamo il seguente problema:

$$m^* = \arg \max_m \ln p(m | z_{1:t}, x_{1:t})$$

Cercare una mappa ottima che massimizza  $p(m | z_{1:t}, x_{1:t})$  equivale a cercare una mappa ottima che massimizza  $\ln p(m | z_{1:t}, x_{1:t})$ . Ricordando la formula di Bayes, vale la relazione:

$$\ln p(m | z_{1:t}, x_{1:t}) = \ln \eta + \ln p(z_{1:t} | x_{1:t}, m) + \ln p(m)$$

Analizziamo i vari termini separatamente.

La costante  $\ln \eta$  può essere trascurata in quanto non influenza il problema di ottimizzazione.

Il termine  $\ln p(z_{1:t} | x_{1:t}, m)$  si decompone in una sommatoria di densità che rappresentano il modello del sensore

$$\ln p(z_{1:t} | x_{1:t}, m) = \ln p(z_1, z_2, \dots, z_t | x_1, x_2, \dots, x_t, m) = \leftarrow z_i \text{ indipendenti tra di loro}$$

$$= \ln p(z_1 | x_1, x_2, \dots, x_t, m) p(z_2 | x_1, x_2, \dots, x_t, m) \dots p(z_t | x_1, x_2, \dots, x_t, m) = \leftarrow Hp \text{ Markoviana}$$

$$= \ln p(z_1 | x_1, m) p(z_2 | x_2, m) \dots p(z_t | x_t, m) =$$

$$= \ln \prod_{i=1}^t p(z_i | x_i, m) = \sum_{i=1}^t \ln p(z_i | x_i, m)$$

Infine  $\ln p(m)$  rappresenta lo stato di occupazione prima che giungano le misure. Avendo considerato una mappa divisa in celle  $p(m)$  è la probabilità di ottenere una determinata sequenza di celle libere e occupate. Denotando con  $p(m_i)$  la probabilità di occupazione di un'unica cella, vale la relazione:

$$p(m) = \prod_{i=1}^N p(m_i)^{m_i} (1 - p(m_i))^{1-m_i} = \text{cost} \prod_{i=1}^N p(m_i)^{m_i} (1 - p(m_i))^{-m_i}$$

Notare che questa relazione vale solo se si assumono le celle indipendenti. Questa ipotesi è totalmente differente da quella fatta nel paragrafo 3.2: qui indichiamo che le celle sono indipendenti prima di considerare la misura (*indipendenza semplice*); nel par.3.2 le celle erano indipendenti dopo aver integrato le misure (*indipendenza condizionale*). Applicando il logaritmo otteniamo:

$$\ln p(m) = \text{cost} + \sum_{i=1}^N m_i \cdot l_0 \quad \text{con } l_0 = \ln \left( \frac{p(m_i)}{1 - p(m_i)} \right)$$

Ricombinando i termini la mappa ottima è ottenibile:

$$m^* = \arg \max_m \sum_{j=1}^t \ln p(z_j | x_j, m) + l_0 \sum_{i=1}^N m_i$$

con  $t$  e  $N$  rispettivamente numero totale di posizioni e celle. La quantità  $\sum_{i=1}^N m_i$  indica

il numero di celle della mappa occupate. Notiamo che se la probabilità di occupazione di una cella  $p(m_i)$  è 0.5,  $l_0 = 0$  e il problema diventa:

$$m^* = \arg \max_m \sum_{j=1}^t \ln p(z_j | x_j, m)$$

Poiché un robot è equipaggiato di solito con  $q$  sensori la relazione precedente diventa

$$(3.1) \quad m^* = \arg \max_m \sum_{i=1}^q \sum_{j=1}^t \ln p(z_{i,j} | x_j, m)$$

Ci rimane da specificare quale modello del sensore usare. Il più utilizzato è quello che rappresenta la misura reale  $z_{i,j}$  (ottenuta dal sensore  $i$ -esimo nella posa  $j$ -esima) come la somma di un termine  $h_i(x_j, m)$  (che indica la lettura ideale del sensore  $i$ -esimo nella posa  $x_j$  riferita alla mappa  $m$ ) e un termine  $v_{i,j}$  (che indica il rumore assunto gaussiano, con media nulla e varianza  $\sigma^2$ ). Il modello del sensore appare dunque nella forma:

$$p(z_j | x_j, m) = p(v_{i,j} = z_{i,j} - h_i(x_j, m))$$

Quanto appena detto ci permette di semplificare l'equazione 3.1. Ricordando la forma della funzione densità gaussiana, deriva che:

$$\ln p(y) = \ln \left( \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{y^2}{2\sigma^2}} \right) = \ln \frac{1}{\sqrt{2\pi\sigma^2}} - \frac{y^2}{2\sigma^2} = c_1 - c_2 \cdot y^2$$

$$\Rightarrow \arg \max_y (c_1 - c_2 \cdot y^2) = \arg \min_y y^2$$

Se  $y = z_{i,j} - h_i(x_j, m)$ ,

$$(3.2) \quad \boxed{m^* = \arg \min_m \sum_{i=1}^q \sum_{j=1}^t (z_{i,j} - h_i(x_j, m))^2 = \arg \min_m \sum_{i=1}^q d_i^2}$$

Questa relazione ci fornisce uno spunto per realizzare un algoritmo di mappatura. Sia  $z_{i,j}$  la misura (ottenuta dall' $i$ -esimo sensore nella posa  $j$ -esima) dell'ambiente reale di

cui vogliamo costruire una mappa. Sia  $h_i(x_j, m)$  la misura (ottenuta dall' $i$ -esimo sensore nella posa  $x_j$ ) di un ambiente  $m$  qualsiasi (anche vuoto). Il principio di partenza è: se una variazione di  $m$  produrrà una diminuzione della quantità  $\sum_{i=1}^q d_i^2$ , tale variazione sarà inglobata nella mappa  $m$  altrimenti verrà scartata e si passerà ad una nuova variazione di  $m$ . Considerando quanto appena detto torna utilissimo l'approccio basato su griglia occupazionale: ogni variazione della mappa corrisponderà ad un cambiamento di stato di una cella. Algoritmi operanti in questo modo vengono detti "Hill Climbing" e verranno discussi nel prossimo paragrafo.

Torniamo sul significato di  $d_i$  indicato nell'espressione precedente. Esso equivale a

$$\begin{aligned} d_i &= \sqrt{(z_{i,1} - h_i(x_1, m))^2 + (z_{i,2} - h_i(x_2, m))^2 + \dots + (z_{i,t} - h_i(x_t, m))^2} \\ &\equiv \left\| (z_{i,j} - h_i(x_j, m)) \right\|_2 \quad j = 1, \dots, t \end{aligned}$$

Ricordando il significato di norma 2, possiamo dire che  $d_i$  rappresenta la distanza dello spazio formato dai punti  $z_{i,j}$  ( $M_{reale}$ ) dallo spazio formato dai punti  $h_i(x_j, m)$  ( $M_{virtuale}$ ). Dunque  $d_i$  misura la diversità tra  $M_{reale}$  e  $M_{virtuale}$ . Pertanto ogni variazione di  $m$  sarà accettata se farà diminuire tale distanza. Per quanto detto la quantità  $\sum_{i=1}^q d_i^2$  rappresenta il distacco della mappa che stiamo costruendo da quella vera e può essere usato come un indicatore di corrispondenza tra mappa reale e mappa virtuale. L'equazione 3.2 può essere riscritta come:

$$m^* = \arg \min_m \sum_{i=1}^q \left\| z_{i,j} - h_i(x_j, m) \right\|_2^2$$

### 3.3.2 Algoritmi “Hill Climbing”

L’approccio usato per la generazione di una mappa rientra in una classe di metodi molto nota nel campo dell’ottimizzazione locale: gli algoritmi “Hill Climbing”. Questo metodo cerca di massimizzare o minimizzare una funzione obiettivo  $f(x)$ . Per fare ciò ad ogni iterazione modifica il valore di un singolo elemento di  $x$  e determina se il cambiamento migliora il valore di  $f(x)$ . Notare che tale tecnica differisce dal metodo del gradiente discendente in quanto quest’ultimo ad ogni iterazione cambia il valore di tutto lo stato  $x$  in accordo con la direzione del gradiente. Inoltre gli algoritmi Hill Climbing non richiedono che la funzione obiettivo sia differenziabile e per questo vengono usati quando tali funzioni sono molto complesse. In più la semplicità implementativa li rende molto popolari nei problemi di ottimizzazione. Sfortunatamente approcci del genere presentano alcuni inconvenienti: risentono molto della forma che assume la funzione obiettivo (soprattutto creste, vicoli e altopiani); sono in grado di trovare solo massimi locali e non globali.

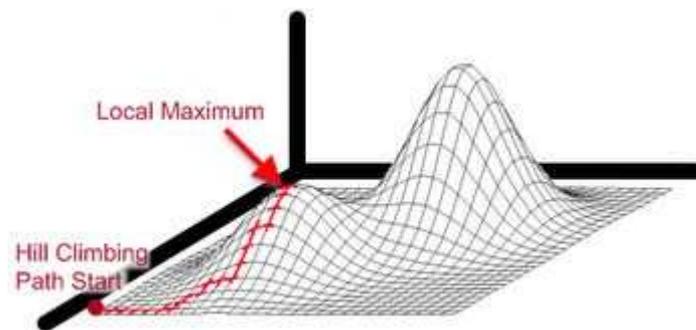


Figura 3.9 Ricerca di un massimo locale: cambiando un elemento del vettore di stato alla volta, in ogni step si muoverà nella direzione del corrispondente asse allineato (il metodo del gradiente discendente può muoversi in tutte le direzioni)

### 3.3.3 Implementazione dell'algoritmo

Indicata la strada da percorrere non ci resta che trovare un algoritmo che la segua. Una soluzione basata sull'approccio Hill Climbing è la seguente:

1. Si divide l'ambiente da mappare in un numero finito di celle di grandezza pre-stabilita.
2. Si inizializzano le celle della mappa  $m$  come libere.
3. Si ispezionano tutte le celle; per ognuna si individua quale stato (1 o 0) minimizza la quantità  $\sum_{i=1}^q d_i^2$  e si assegna quello stato alla mappa  $m$ .
4. Si ripete fino alla convergenza.

Mostriamo con un esempio come lavora questo algoritmo. Consideriamo l'ambiente mostrato in fig.3.10: un robot dotato di un sensore occupa le posizioni  $x_1, x_2, x_3$  e effettua le corrispondenti misurazioni  $z_1, z_2, z_3$ .

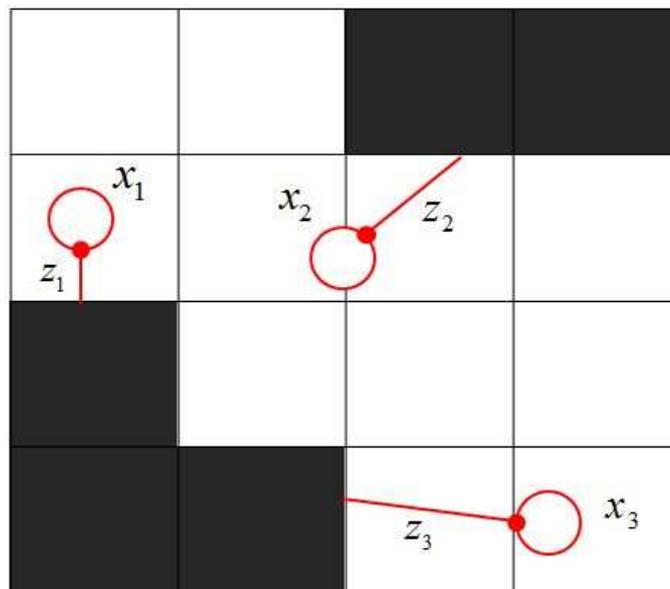


Figura 3.10 Rappresentazione dell'ambiente in griglie di occupazione

L'avanzamento dell'algoritmo riferito alle prime due celle è mostrato nella fig. 3.11.

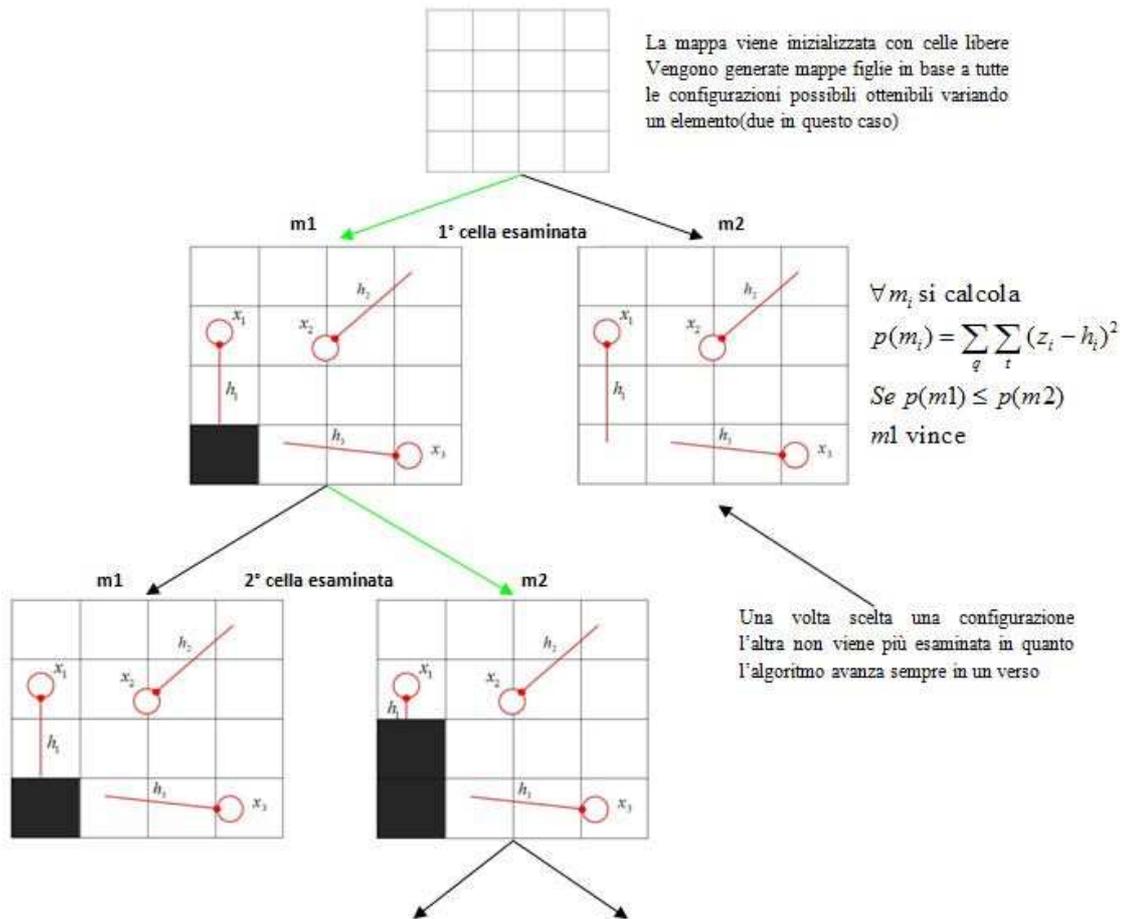


Figura 3.11 Algoritmo di mappatura basato su stimatore MAP

Le celle vengono tutte esaminate e alla fine si confronta la mappa ottenuta con la mappa di partenza. Se le due mappe coincidono l'algoritmo ha raggiunto la mappa di convergenza, altrimenti si ripete l'ispezione di tutte le celle.

Come tutti gli algoritmi hill climbing, questo algoritmo trova solo i massimi locali, ma la sua efficienza è garantita dal fatto che la funzione  $p(m | z_{1:t}, x_{1:t})$  presenta un numero ridotto di massimi. Inoltre essendo un approccio che ottimizza la distribuzione di

probabilità, non tiene conto dell'incertezza residua nella rappresentazione della mappa. Questo tipo di metodo offre risultati nettamente migliori nella ricostruzione dell'ambiente: infatti esso non richiede un numero molto elevato di osservazioni come invece richiedeva l'algoritmo basato su filtro binario. Applichiamo questo metodo all'esempio rappresentato in fig.3.8. La mappa costruita dal filtro binario non riusciva a rappresentare la porta aperta (fig. 3.8b). La fig.3.12 mostra la mappa derivante dall'applicazione dell'algoritmo MAP: la porta aperta è ben rappresentata anche se il numero di misure è ristretto.

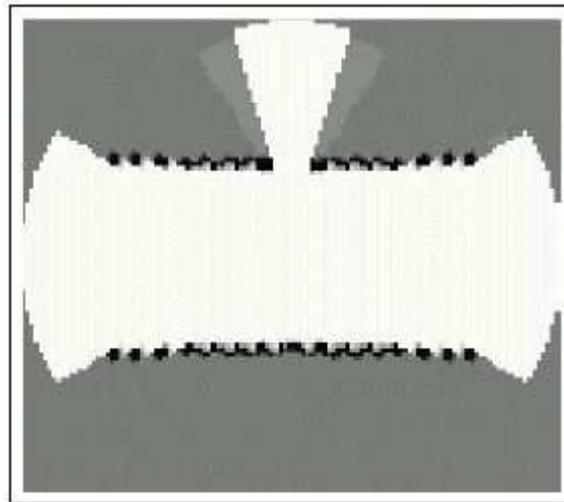


Figura 3.12 Mappa basata sullo stimatore MAP [1]

Un altro svantaggio di questo algoritmo è l'incremento della complessità computazionale, che può essere ridotta inizializzando la mappa di partenza non a zero ma con il risultato dell'algoritmo di mappatura basato su filtro binario.

## 3.4 Tecniche Alternative

### 3.4.1 Mappatura basata sulla Probabilità di Riflessione

Nei paragrafi precedenti abbiamo visto che l'approccio basato sul calcolo della mappa ottima offre risultati migliori dal punto di vista della rappresentazione, rendendo le mappe adatte per essere utilizzate nella navigazione. Nel paragrafo 3.3.1 abbiamo fornito un metodo per la costruzione dello stimatore che si basa sul modello del sensore. Un metodo alternativo a questo viene conosciuto con il nome di *Mappatura basata sulla probabilità di riflessione (Reflection Probability Mapping)*. È una tecnica che prevede sempre la ricerca della mappa ottima

$$m^* = \arg \max_m p(m | z_{1:t}, x_{1:t})$$

ma che la calcola considerando la probabilità di riflessione di una cella, ovvero la probabilità che un raggio del sensore sia riflesso dalla cella. Dopo una serie di passaggi otteniamo che la mappa più probabile è data da:

$$m^* = \arg \max_m \sum_{i=1}^N (\alpha_i \ln m_i + \beta_i \ln(1 - m_i))$$

Dove  $\alpha_i$  corrisponde col numero di volte che un fascio viene riflesso dalla cella  $i$ -esima,  $\beta_i$  coincide al numero di volte che un fascio attraversa la cella  $i$ -esima e

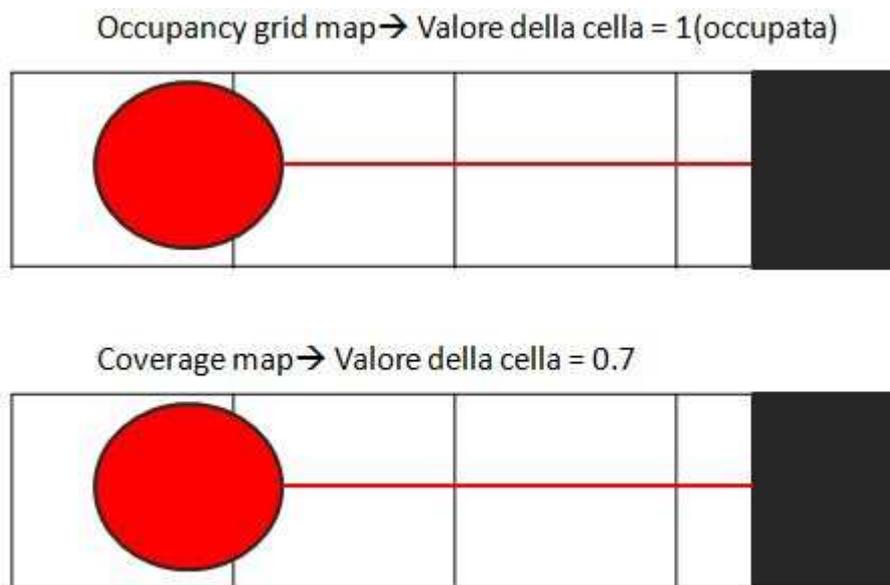
$m_i = \frac{\alpha_i}{\alpha_i + \beta_i}$ , che equivale proprio alla probabilità di riflessione della cella  $m_i$ .

Pertanto il calcolo della mappa più probabile si riduce al calcolo di quanto spesso una

cella riflette o intercetta un fascio. Notiamo infine che mentre nell'approccio basato su filtro binario i valori di probabilità di ogni cella convergevano verso gli estremi [0-1], qui convergono verso il centro dell'intervallo. Per ulteriori dettagli vedi [1].

### 3.4.2 Mappatura basata sulla probabilità di copertura

Questa tecnica di mappatura si basa anch'essa sulla divisione dell'ambiente in celle, ma per ogni cella memorizza la percentuale di occupazione (*Coverage Map*). Negli algoritmi presentati le celle venivano considerate o libere o occupate, in questo caso si realizza una distribuzione di probabilità per ogni cella per indicarne la percentuale di copertura. Per ulteriori informazioni vedi [8].





# Capitolo 4

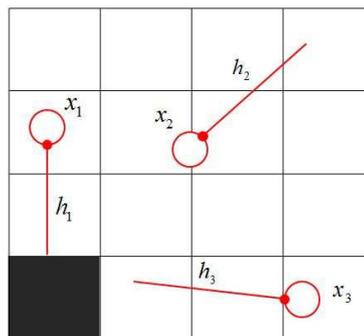
## Simulazioni

### 4.1 Introduzione

Nei paragrafi precedenti abbiamo introdotto due tecniche di mappatura fondate entrambe sulla scomposizione dell'ambiente in celle: la prima utilizza il filtro binario di Bayes e assegna ad ognuna un valore che ne indica lo stato di occupazione (tale valore espresso nella forma log odds ratio tende velocemente a 1 o 0); la seconda cerca tra tutte le possibili mappe quella più probabile e viene realizzata attraverso uno stimatore MAP implementato con un algoritmo di tipo "Hill Climbing". Abbiamo visto che la prima tecnica, introducendo l'ipotesi di indipendenza condizionata delle celle, genera una mappa la cui precisione dipende dal numero di osservazioni. La seconda invece produce mappe più accurate a discapito di un leggero aumento della complessità computazionale. In questo capitolo applicheremo le varie metodologie ad un ambiente virtuale ottenuto mediante un software simulativo e confronteremo i risultati dei vari metodi.

## 4.2 L'ambiente, il robot, il sensore

Il robot è stato rappresentato come un oggetto di forma circolare di raggio  $d1$ . Nel corso delle prove si è assunto  $d1=0$ . Il robot è dotato di otto sensori posti a  $45^\circ$  l'uno dall'altro. Il fascio di ogni sensore è stato rappresentato come un segmento uscente dal robot di lunghezza  $z\_max$ . Il valore di  $z\_max$  è stato spesso modificato. I risultati rappresentati in questo capitolo sono stati ottenuti con  $z\_max=2$  metri. Il rumore, aggiunto alle misure, è stato assunto gaussiano con media nulla e varianza  $\sigma^2 = 0.02 \text{ m}^2$ . Nell'implementazione degli algoritmi si è tenuto conto di un aspetto importante: conoscendo a priori le posizioni assunte dal robot, sappiamo con certezza che le celle corrispondenti a tali posizioni sono libere.



Un punto cruciale dell'algoritmo è l'implementazione della funzione per il calcolo di  $h_i(x_j | m)$ , ovvero della distanza misurata dal sensore  $i$ -esimo nella posizione  $j$ -esima nell'ambiente che stiamo costruendo. Considerando il suo compito, essa va ripetuta per tutte le celle per ogni iterazione fino alla convergenza. Nella figura 4.1 viene mostrata in rosso la cella che stiamo valutando. Le cifre rappresentano invece il sensore più vicino alla cella, calcolato con la procedura mostrata nel par.3.2. Un metodo per ridurre il numero di iterazioni è il seguente: definita la cella che sta cambiando valore, si calcola la distanza  $h_i(x_j | m)$  solo per il gruppo di celle che ha lo stesso numero di sensore. Considerando la figura, se la cella che cambia è quella rossa, le celle da valutare saranno solo quelle etichettate con il valore 4. Definito ciò la distanza è calcolata

semplicemente come l'intersezione tra il segmento del sensore e le pareti della cella. I miglioramenti apportabili sono numerosi, ma come prima implementazione risulta abbastanza efficiente.

4	4	3	2
4	4	3	2
5	5	●	1
6	6	7	8

Figura 4.1

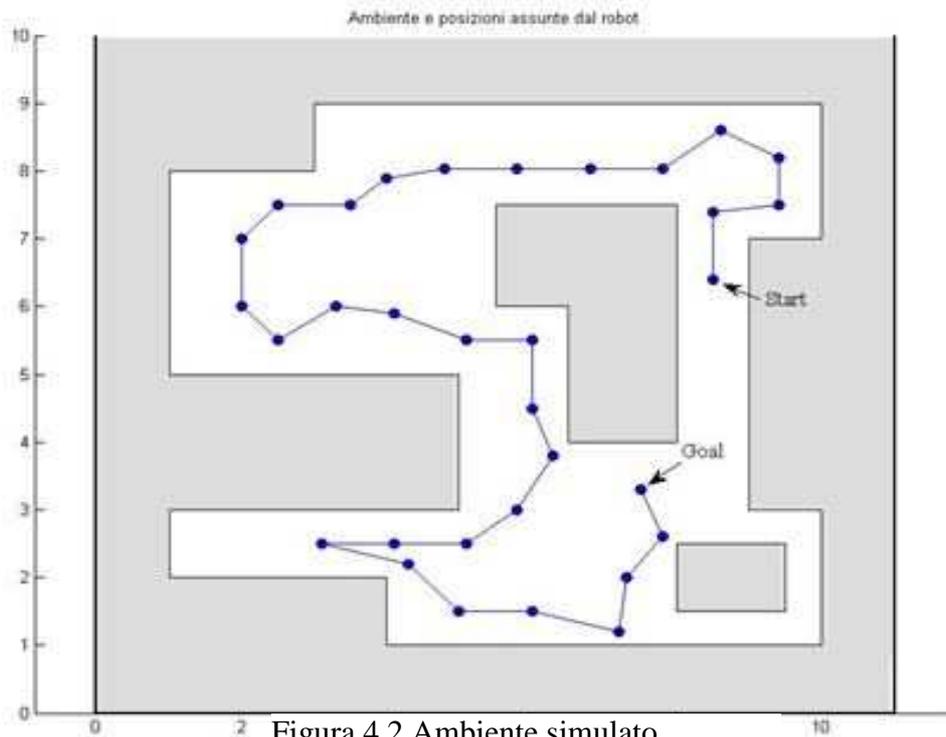
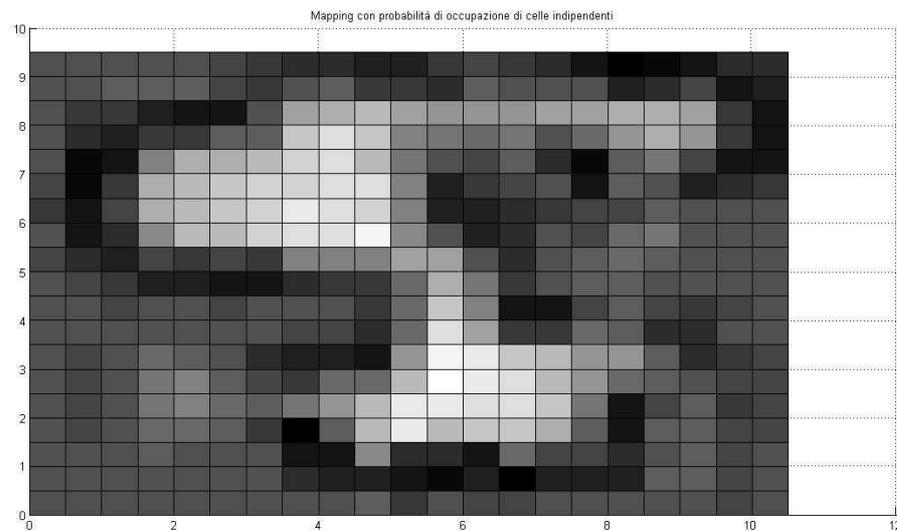


Figura 4.2 Ambiente simulato

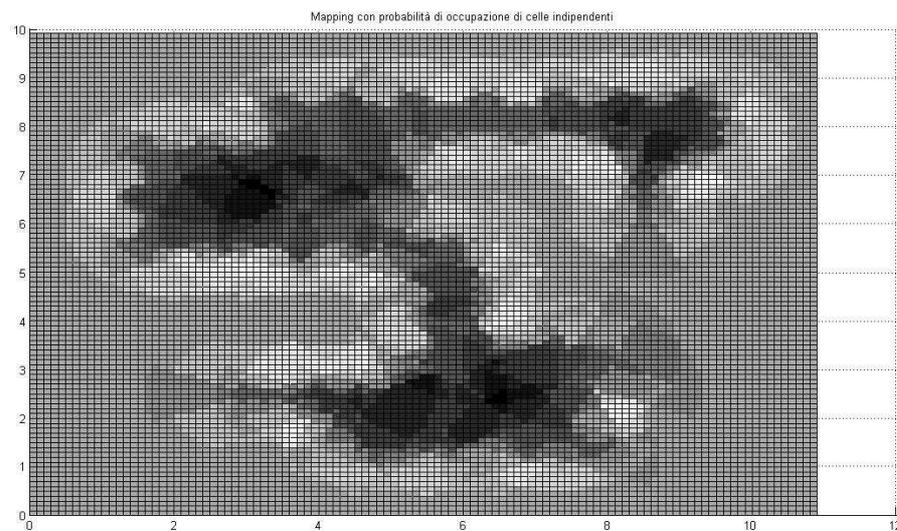
La figura 4.2 mostra l'ambiente sul quale verranno testati gli algoritmi. Il robot (il punto blu) percorre una traiettoria costituita dalla sequenza di varie posizioni da un punto iniziale (start) ad uno finale (goal).

## 4.3 Simulazioni

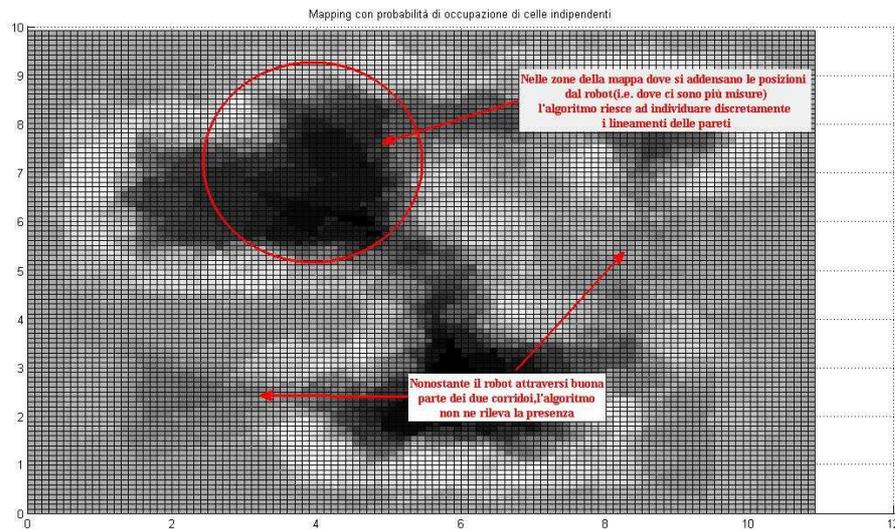
L'immagine in basso mostra la mappa ottenuta dall'applicazione della tecnica basata su filtro binario con celle di grandezza 0.5. Il risultato è completamente insoddisfacente e ci costringe a ridurre la dimensione delle celle.



L'immagine seguente mostra lo stesso algoritmo applicato a celle di grandezza 0.1.



I risultati sono nettamente migliorati. Questa simulazione mostra tutti i difetti di questo approccio, evidenziati nell'immagine che segue.



Come si può notare (la parte evidenziata dal cerchio) la rappresentazione inizia a diventare buona nelle zone dove si addensano le posizioni e quindi le misurazioni sono maggiori (forte dipendenza dal numero di posizioni assunte). Il secondo difetto, che rappresenta la limitazione maggiore, è evidenziato dalle due frecce in basso: nonostante il robot attraversi gran parte dei due corridoi, questi non vengono identificati nella mappa. Tale approccio è dunque inadatto per pianificare un percorso di navigazione. La figura 4.3 mostra l'applicazione dell'algoritmo di mapping basato sulla ricerca della mappa ottima. Come è facile notare tale mappa si presta molto bene per la navigazione di un robot. Il valore in alto indica quanto la mappa che stiamo costruendo si avvicina a quella reale. Vengono presentati alcuni risultati ottenuti variando l'ampiezza delle celle: la figura 4.4 mostra una mappa ottenuta con celle di grandezza 0.4; la figura 4.5 mostra una mappa ottenuta con celle di grandezza 0.3.

Rappresentazione della mappa con celle di grandezza 0.5.

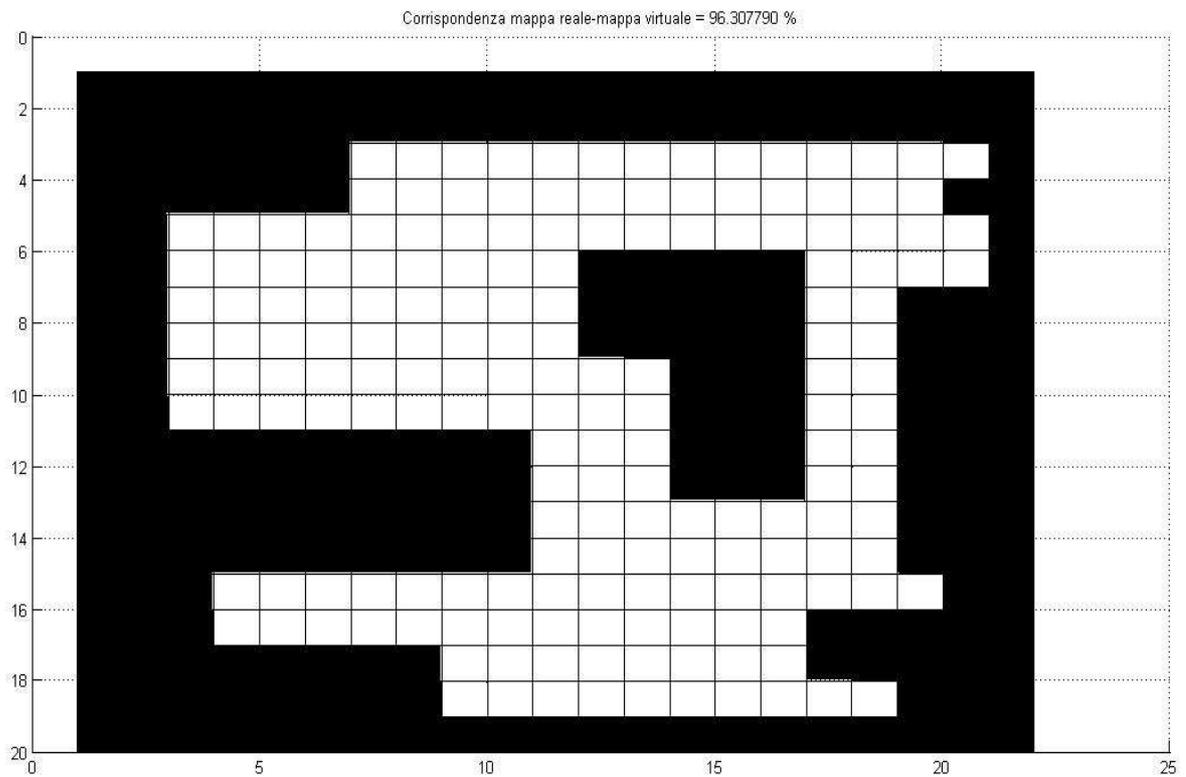


Figura 4.3 Mappa ottenuta attraverso l'algoritmo MAP (Maximum A Posteriori) con celle di grandezza 0.5



Rappresentazione della mappa con celle di grandezza 0.3.

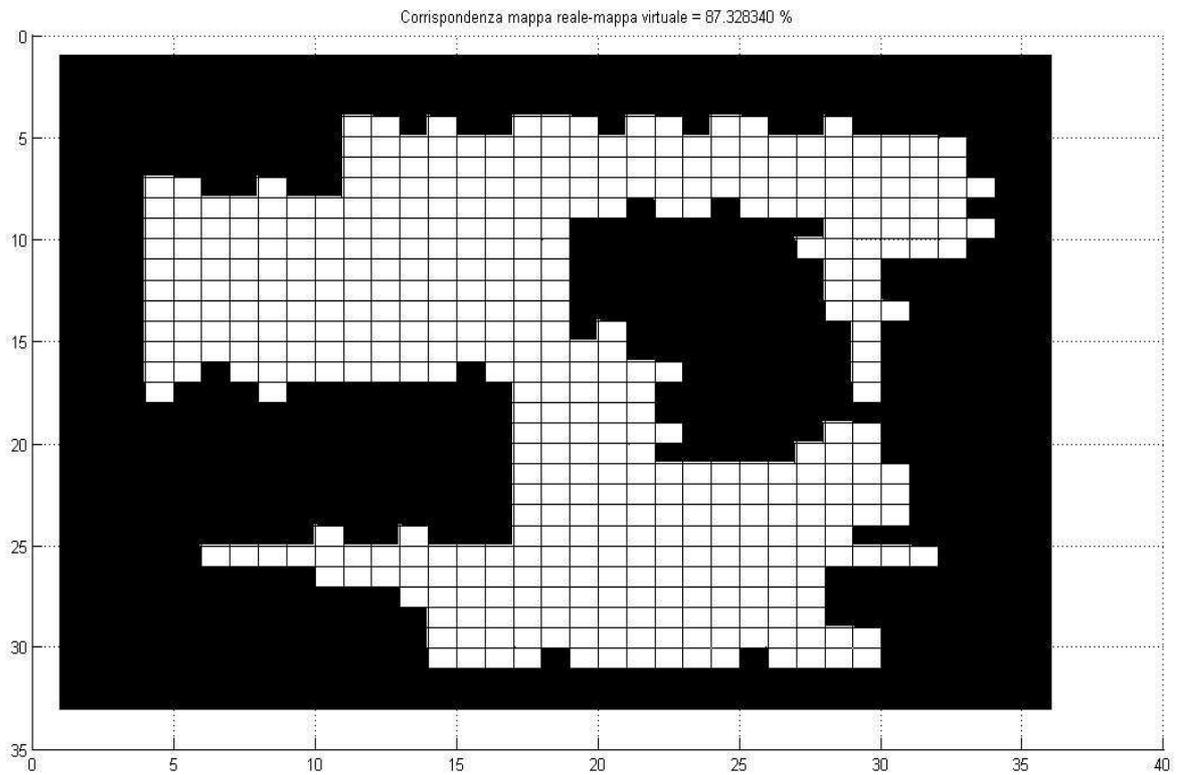
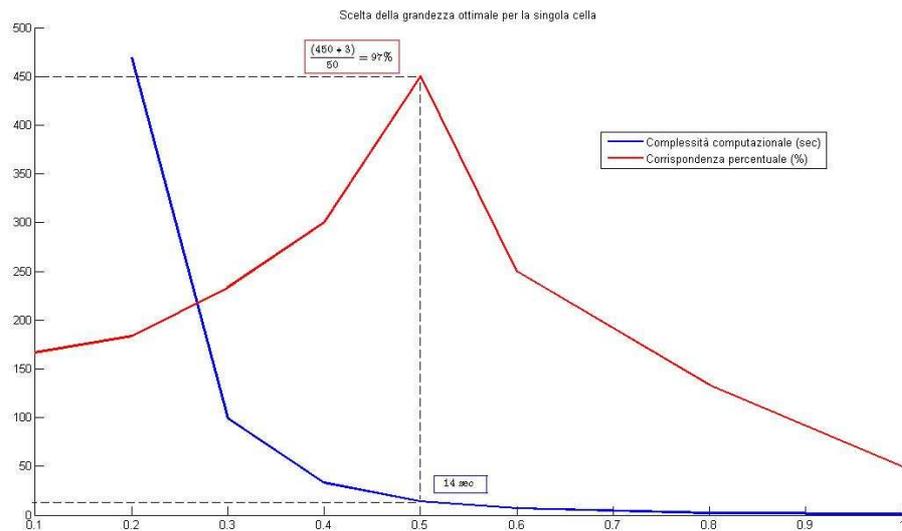


Figura 4.5 Mappa ottenuta attraverso l'algoritmo MAP (Maximum A Posteriori) con celle di grandezza 0.3

Il grafico in basso mostra il tempo di esecuzione dell'algoritmo (blu) e la corrispondenza percentuale tra mappa reale e virtuale (rosso) in funzione della grandezza delle celle.



Notiamo che il miglior rapporto corrispondenza-tempo di esecuzione si ha con celle di grandezza 0.5. Infine per quanto riguarda la complessità computazionale dell'algoritmo, possiamo affermare che essa cresce linearmente col numero di posizioni, con l'area da mappare e con il numero di iterazioni necessarie a raggiungere la convergenza.

## 4.4 Conclusioni e Sviluppi futuri

In questo lavoro abbiamo sviluppato algoritmi adatti alla mappatura di ambienti per la navigazione di robot mobili. Ne abbiamo fornito due soluzioni: la prima si basava sull'applicazione del filtro binario di Bayes, il quale aggiornava in modo iterativo il valore delle singole celle della mappa. La rappresentazione risultante presenta alcune limitazioni. Innanzitutto quando il sensore rileva un ostacolo tutte le celle dell'arco del cono del sensore vengono etichettate come occupate, producendo condizioni di occupazione errati. Inoltre si è notato che le celle osservate più frequentemente tendono ad assumere valori agli estremi dell'intervallo  $(1,0)$ . Quindi la precisione della mappa è fortemente dipendente dal numero di osservazioni. Questa caratteristica si riflette nell'impossibilità di rappresentare luoghi attraversati velocemente dal robot dove il numero di osservazioni è ridotto. Tale mappa si presta male alla navigazione in quanto non si distinguono le celle occupate da quelle libere. Una soluzione potrebbe essere quella di applicare una sogliatura alla mappa risultante compromettendo però la precisione della rappresentazione. Tutte queste limitazioni vengono superate dall'approccio basato su uno stimatore Maximum A Posteriori che realizza mappe mantenendo la condizione di dipendenza condizionale delle celle e la cui accuratezza è meno sensibile al numero di osservazioni. Queste mappe sono adatte ad essere usate nella pianificazione del percorso in quanto si evidenziano con cura le celle libere. La precisione di queste mappa è pagata con un aumento della complessità computazionale dell'algoritmo. Una soluzione è l'utilizzo congiunto dei due metodi: la ricerca della mappa ottima non parte più da una mappa vuota, ma riceve come input la mappa risultante dall'applicazione del filtro binario. Inoltre si può pensare di rendere variabile la grandezza delle celle in modo tale che ambienti ricchi di caratteristiche siano rappresentati con celle più piccole aumentando così la precisione della mappa. L'utilizzo della mappa ottima rende semplice la ricerca di un percorso. Infatti si può pensare di trasformare la rappresentazione metrica in una topologica dove ogni nodo rappresenta

una cella libera o gruppi di celle adiacenti libere ed esiste un collegamento tra due nodi se esiste una cella libera che li unisce. In questo modo individuare il giusto percorso si trasforma nel problema di ricerca di un cammino minimo in un grafo, risolvibile con i vari metodi evidenziati dalla Teoria dei Grafi.



# Bibliografia

- [1] Thrun, Burgard, Fox, *Probabilistic Robotics*, MIT Press
- [2] F. Martinelli, appunti del corso " Mobile Robot Localization", Università di Roma Tor Vergata , <http://robot2.disp.uniroma2.it/fmartine/FiltRob/>
- [3] ISTC-CNR, "RobotCare Project", <http://robocare.istc.cnr.it/>
- [4] ISTC-CNR, "RobotCup Rescue", <http://www.istc.cnr.it/>
- [5] M. Mazzeo, "Tecniche di mappatura per la navigazione di robot mobili" , tesi, Università di Roma "Tor Vergata", 2007/2008
- [6] G. Tipaldi, "Modelli approssimati per la localizzazione e mapping mediante robot mobili", tesi, Università di Roma "La Sapienza", 2004/2005
- [7] "Mapping & Localization", <http://www.irobot.net/>
- [8] Cyrill Stachniss, "Exploration and mapping with mobile robots", Ph.D. thesis, University of Freiburg, 2006
- [9] F. Ramundo, "Metodi information-based di esplorazione robotica", tesi, Università di Roma "La Sapienza", 2005/2006