

TITOLO DA INSERIRE

FRANCESCO ARGENTIERI*

GIACOMO MAZZAGLIA†

Università di Trento

Università di Trento

francesco.argentieri@studenti.unitn.it

giacomo.mazzaglia@studenti.unitn.it

11 maggio 2018

Sommario

In questo lavoro, consideriamo il problema di esplorare un ambiente sconosciuto con un team di robot. Come nell'esplorazione di robot singoli, l'obiettivo è di ridurre al minimo il tempo di esplorazione complessivo. Il problema chiave da risolvere nel contesto di robot multipli è quello di scegliere i punti di destinazione appropriati per i singoli robot in modo che possano esplorare contemporaneamente diverse regioni dell'ambiente. Presentiamo un approccio per il coordinamento di più robot, che tiene conto simultaneamente del costo di raggiungere un punto target e della sua utilità. Ogni volta che un punto target viene assegnato a un robot specifico, l'utilità dell'area inesplorata visibile da questa posizione target viene ridotta. In questo modo, le diverse posizioni di destinazione vengono assegnate ai singoli robot. Descriviamo inoltre come il nostro algoritmo può essere esteso a situazioni in cui il raggio di comunicazione dei robot è limitato. Per la stima delle posizioni dei robot è stato utilizzato il filtro particellare, assumendo una comunicazione con delle antenne wi-fi. I risultati dimostrano che la nostra tecnica distribuisce efficacemente i robot sull'ambiente e consente loro di compiere rapidamente la loro missione.

*ID: 183892

†ID: 183382

I. INTRODUZIONE

L'esplorazione efficiente di ambienti sconosciuti è un problema fondamentale nella robotica mobile. L'estensione alla esplorazione di più robot pone diverse nuove sfide, tra cui: coordinamento di robot, integrazione delle informazioni raccolte dai robot in una mappa coerente; e comunicazione limitata. Il coordinamento di un sistema di più robot è la base per un'efficiente implementazione di una esplorazione distribuita. Difficoltà maggiore nel campo del coordinamento vengono dalle conoscenze che si assumono avere i robot sul comportamento degli altri robot. Se i robot conoscono le loro posizioni relative e condividono una mappa della zona che hanno esplorato finora, si può raggiungere un coordinamento efficace, guidando i robot in zone inesplorate dell'ambiente. Questo può essere fatto assegnando ai robot il compito di raggiungere un loro punto di arrivo preso dalla frontiera fornita dalla scansione del lidar(paper). Per un assegnamento efficace è importante che i robot quando vadano a condividere dei punti comuni all'interno delle loro frontiere che non cerchino di raggiungere lo stesso punto. Per la stima della posizione viene adottato un filtro particellare, i robot comunicando con delle antenne wi-fi riescono a conoscere con un dato errore la loro posizione ricoperta in quell'istante. Per il raggiungimento della zona target prefissata viene invece adottata la funzione potenziale. I robot contengono al loro interno memoria delle zone già esplorate e non, ipotizzando di conoscere in anticipo le dimensioni massime della mappa che andranno ad esplorare quest'ultimi costruiscono attraverso l'integrazione di successive matrici di occupazione locale una matrice di occupazione globale che rappresenta la mappatura dell'ambiente. Saranno successivamente presentati la modellazione del sistema e i successivi risultati ottenuti

II. COMUNICAZIONE

Esistono 2 tipi di comunicazione all'interno del nostro sistema. Un primo tipo di comunicazione è quella disponibile tra robot e che

serve per l'assegnazione e coordinazione efficiente del sistema composto da più robot, la comunicazione tra robot che verrà considerata è una comunicazione ideale senza perdita di informazione nel quale i robot possono comunicarsi le loro posizioni stimate tramite particle filter e la mappa esplorata fino a quel momento dal robot. Un'altro tipo di comunicazione è quello disponibile tra robot ed antenne wi-fi disposte casualmente o tramite input dell'utente all'interno della mappa e che permettono la stima della posizione coperta dal robot in quel dato istante, anche questa viene assunta essere una comunicazione ideale senza perdita di informazioni.

III. MODELLO DEL SISTEMA

i. Modello cinematico

Il robot è basato sul modello dell'uniciclo a trazione differenziale, la configurazione è completamente descritta da $\mathbf{q} = [x \ y \ \theta]^T$, dove (x, y) sono le coordinate cartesiane del punto di contatto con il suolo e θ è l'orientamento della ruota rispetto l'asse x . [1], come in figura 3. Il modello cinematico dell'uniciclo è descritto dall'equazioni (1):

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos \theta \\ \sin \theta \\ 0 \end{bmatrix} v + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \omega \quad (1)$$

Il robot ha le dimensioni riportate in tabella 1.

Tabella 1: Riepilogo dimensioni

dimensioni		
raggio ruote	[m]	0.07
interasse	[m]	0.30

Questo è equipaggiato con un sensore virtuale *lidar*, basato sul modello Hokuyo URG-04LX, collocato al centro della struttura in modo tale da evitare errori di offset, di seguito se ne riportano le caratteristiche, di cui adatte ad hoc per la simulazione. Come sensori propriocettivi presenta due encoder incrementali virtuali calettati sull'asse delle ruote, le caratteristiche di entrambi sono riportate in

tabella 2. Una rappresentazione del robot è osservabile in figura 3.

Tabella 2: Specifiche sensori

specifiche lidar virtuale		
risoluzione angolare	$[\circ]$	0.36
angolo di scansione	$[\circ]$	180.00
massima distanza	$[m]$	4.00
minima distanza	$[m]$	0.02
risoluzione	$[mm]$	1.00
specifiche encoder virtuale		
risoluzione	$2 \cdot (\frac{\pi}{2600})$	

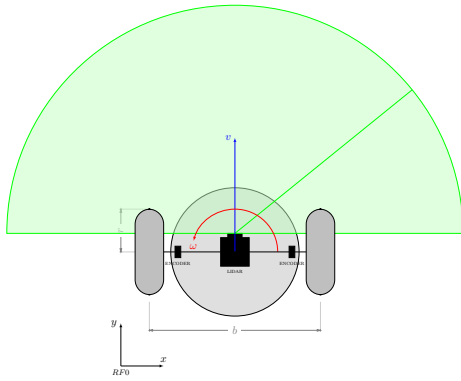


Figura 1: modello cinematico

ii. Modello del sensore ed Occupancy grid

Gli approcci basati sulle occupancy grid tendono a non ragionare sulla geometria delle caratteristiche dell'ambiente, ma ad assumere una visione più sensoriale. Una rete di celle (una griglia) è posta sopra il (inizialmente sconosciuto) mondo. Ogni cella $c(i; j)$ ha un numero associato ad essa corrispondente a una probabilità di occupazione - $po(i; j)$ e una probabilità di essere vuota pe . Non facciamo affermazioni in merito ciò che sta occupando solo che qualcosa o parte di qualcosa è alle coordinate coperte per cella (40,20).

ii.1 Occupancy grid locale

Per un determinato sensore di rilevamento della distanza, come uno scanner per linee laser, con portata massima R e mezza larghezza del

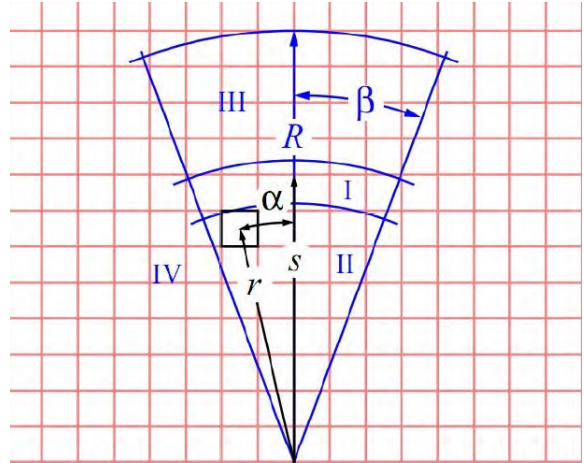


Figura 2: Scan scheme

raggio del sensore β , il modello può essere scomposto in un numero di settori etichettati I-IV, come illustrato nella Figura. .

La regione I è la regione associata alla lettura effettiva del laser, la regione II rappresenta un'area vuota in cui nulla viene rilevato, la regione III è l'area coperta dal raggio laser, ma rimane sconosciuto se è occupata o meno a causa dell'occlusione, e la regione IV è al di fuori del campo di visibilità del Lidar.

Parametri rilevanti per il ragionamento sulla cella evidenziata (contorno nero) includere r , che è la distanza dell'elemento di griglia dalla posizione del sensore, e α , l'angolo dell'elemento di griglia relativo al raggio centrale del sensore.

Per una misurazione s che ricade nella regione I, la probabilità che la misura fosse dovuta a un ostacolo effettivo presente a tale intervallo e il suo complemento può essere calcolato da:

$$P(s|Occupied) = \frac{\frac{R-r}{R} + \frac{\beta-\alpha}{\beta}\beta}{2} * Max_{occupied} \quad (2)$$

dove $Max_{occupato}$ è dovuto all'assunzione che una lettura di "Occupato" non è mai completamente affidabile. A causa dell'occlusione, assumiamo che la probabilità di occupazione nella regione III sia del 50% occupato e il 50 % non occupato.

ii.2 Occupancy grid globale

Ogni robot presenta al suo interno una memoria interna che tiene conto di una possibile mappatura globale dell'intero ambiente. Questa è definita occupancy grid globale e viene ricostruita rototraslando le varie occupancy grid locali fornite in diversi istanti dal robot.

Un problema con i metodi basati sulla griglia è l'utilizzo della memoria: diventa costoso per il 3D mappe. Un secondo problema è quello della griglia allineamento - è arbitrario e in quanto tale è probabile che le celle coprano aree che sono difficili da raggiungere parzialmente pieno come tale non è possibile catturare le acute discontinuità ai bordi di oggetti. Forse il più grande problema dei metodi basati sulla griglia deriva dalle difficoltà in localizzazione rispetto a una griglia.

IV. SOLUZIONE PROPOSTA

i. Allocazione destinazioni ed Utility

L'obiettivo di un processo di esplorazione è quello di coprire l'intero ambiente nel minor tempo possibile. Pertanto, è essenziale che i robot tengono traccia di quali aree dell'ambiente sono già state esplorate. Inoltre, i robot devono costruire una mappa globale per pianificare i loro percorsi e coordinare le loro azioni.

Quando esploriamo un ambiente sconosciuto, siamo particolarmente interessato alle "celle di frontiera". Con cella di frontiera, denotiamo ciascuna cella già esplorata che è un vicino immediato di una cella sconosciuta, inesplorata. Se indirizziamo un robot a una tale cella, ci si può aspettare che ottenga informazioni sull'area inesplorata quando arriva alla sua posizione di destinazione. Il fatto che una mappa in generale contiene diverse aree inesplorate solleva il problema di come assegnare compiti di esplorazione rappresentati da celle di frontiera ai robot. Se sono coinvolti più robot, vogliamo evitare che molti di loro si spostino nella stessa posizione. Per affrontare questi problemi e determinare i luoghi di destinazione appropriati per i singoli robot il nostro sistema utilizza il seguente approccio: consideriamo contemporaneamente il costo del

raggiungimento di una cella di frontiera e l'utilità di quella cella. Per ogni robot, il costo di una cella è proporzionale alla distanza tra il robot e quella cella e l'utilità di una cella di frontiera dipende invece dal numero di robot che si stanno spostando in quella cella o in un posto vicino a quella cella. Di seguito vengono riportate le equazioni implementate

$$U(t_n|t_1, \dots, t_{n-1}) = U_{t_n} - \sum_{i=1}^{n-1} P(||t_n - t_i||) \quad (3)$$

$$P(d) = \begin{cases} 1 - \frac{d}{max_{range}} & \text{if } d < max_{range} \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

$$(i, t) = argmax_{i', t'} (U_{t'} - \beta V_{t'}^{i'}) \quad (5)$$

La prime due equazioni rappresenta come viene calcolata la l'utility per il robot n-esimo mentre l'ultima equazione rappresenta il caso allargato all'integrazione di più robot che comunichino tra loro. Maxrange rappresenta la massima visibilità fornita dal lidar, d rappresenta la distanza dei target assegnati ai vari robot la U la funzione di utility e la V la funzione di costo che tiene conto della distanza di ogni robot dal suo target. Le i e t risultanti dall'ultima equazione forniscono per ogni robot il suo target i-esimo ottimo per avere l'ottimo globale della funzione tra le parentesi. Il modo in cui questi target da parte di ogni robot vengono raggiunti è lasciato alla funzione potenziale che verrà descritta in seguito.

ii. Pianificazione

La pianificazione del percorso per i robot è uno dei criteri importanti da prendere in considerazione per migliorare il livello di autonomia del robot. Nella pianificazione del percorso, la sicurezza è un problema importante che dovrebbe essere preso in considerazione al fine di garantire che un robot raggiunga la posizione target senza collisione con gli ostacoli circostanti. Inoltre, ci sono aspetti importanti che devono essere affrontati nella pianificazione del

percorso; tempo computazionale, percorso ottimale e completezza. Uno dei metodi più diffusi per la pianificazione dei percorsi è il metodo *Campi Potenziali Artificiali*. Il metodo del potenziale è in grado di superare uno scenario sconosciuto, tenendo conto della realtà dell'ambiente corrente del movimento del robot. Due tipi di forze coinvolte nel potenziale metodo del campo; forza attrattiva generata da obiettivi e forza repulsiva generata da ostacoli, di conseguenza, il robot deve riprogrammare un nuovo percorso[2]. Utilizzando informazioni parziali sullo spazio di lavoro raccolte attraverso i sensori quindi le informazioni sensoriali integrate in una mappa secondo un paradigma *sense—plan—move*. Oppure utilizzare le informazioni sensoriali impiegate per pianificare moti secondo un paradigma *stimulus—response* (navigazione reattiva). Il robot è considerato come un punto sotto l'influenza dei campi prodotti da obiettivi e ostacoli nello spazio di ricerca. Le forze repulsive sono generate da ostacoli mentre la forza attrattiva è generata dagli obiettivi. Le forze risultanti (la somma di tutte le forze) dei campi sul robot vengono utilizzate per determinare la direzione del movimento del robot e la velocità di spostamento evitando collisioni[3]. Tuttavia esistono svantaggi quali: **a)** situazione di stallo dovuta ai minimi locali; **b)** oscillazione in presenza di ostacoli; **c)** nessun passaggio tra ostacoli ravvicinati; **d)** oscillazioni in passaggi stretti[4]. Il robot viene considerato come punto $\mathbf{q} = (x\ y)^T$, in un piano cartesiano, attratto (potenziale U_{att}) dal punto obiettivo \mathbf{q}_g e respinto (potenziale U_{rep}) dagli ostacoli.

$$U(q) = U_{att}(q) + U_{rep}(q) \quad (6)$$

dove $U(q)$ potenziale artificiale; $U_{att}(q)$ campo attrattivo; $U_{rep}(q)$ campo repulsivo. La pianificazione avviene in modo incrementale: a ogni configurazione \mathbf{q} , la forza artificiale viene generata come (7)

$$\begin{aligned} F(q) &= -\nabla U(q) \\ &= -\nabla U_{att}(q) - U_{rep}(q) \\ F(q) &= F_{att}(q) + F_{rep}(q) \end{aligned} \quad (7)$$

dove $F(q)$: forza artificiale; $F_{att}(q)$: forza attrattiva; $F_{rep}(q)$: forza repulsiva. Il campo

potenziale U_{att} tra robot e obiettivo viene descritto da (8) per trascinare il robot nell'area obiettivo.

$$\begin{aligned} U_{att}(q) &= \frac{1}{2} k_a (q - q_d)^2 \\ &= \frac{1}{2} k_a \rho_{goal}^2(q) \end{aligned} \quad (8)$$

dove k_a : coefficiente positivo per APF¹; q : posizione corrente del robot; q_d : posizione corrente dell'obiettivo. $\rho_{goal}(q) = \|q - q_d\|$ è una distanza euclidea dalla posizione del robot alla posizione dell'obiettivo. La forza attrattiva del robot è calcolata come gradiente negativo del potenziale campo[5]:

$$\begin{aligned} F_{att}(q) &= -\frac{1}{2} k_a \rho_{goal}^2(q) \\ F_{att}(q) &= -k_a (q - q_d) \end{aligned} \quad (9)$$

$F_{att}(q)$, nell'eq. (9), è un vettore diretto verso q_d con intensità linearmente proporzionale alla distanza da q a q_d . Può essere scritto nelle sue componenti:

$$\begin{aligned} F_{att} - x(q) &= -k_a (x - x_d) \\ F_{att} - y(q) &= -k_a (y - y_d) \end{aligned} \quad (10)$$

Le equazioni (10) sono la forza attrattiva nelle direzioni x e y . Nella funzione potenziale, il robot deve essere respinto dagli ostacoli, ma se lontano da questi, il movimento non risente della loro influenza. La funzione potenziale di repulsione (11) è:

$$U_{rep}(q) = \begin{cases} \frac{1}{2} k_b \left(\frac{1}{d(q)} - \frac{1}{d_0} \right)^2 & \text{se } d(q) \leq d_0 \\ 0 & \text{se } d(q) > d_0 \end{cases} \quad (11)$$

iii. Algoritmi di stima

Il sistema robotico preso in esame presenta degli errori all'ingresso degli encoder, tale errore porta il sistema a manifestare nel lungo periodo il noto comportamento di drift. Nasce allora il problema di correggere e stimare la posizione in base ad un dato modello dello stato presentato già precedentemente il modello odometrico e un modello di sensore esterno al

¹Artificial Potential Field

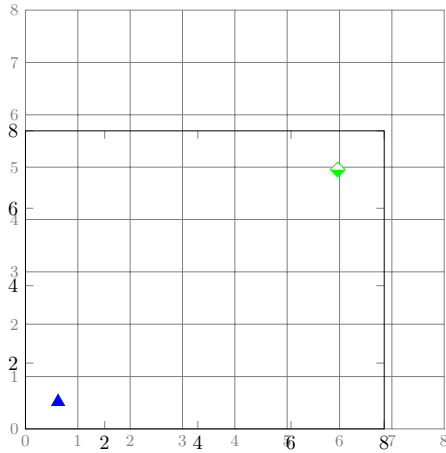


Figura 3: Modello cinematico

robot che fornisca una misura approssimata della sua posizione. I metodi presentati saranno 2, un primo di semplice applicazione ipotizza il ricevimento da parte del robot di un segnale GPS che viene integrato e sfruttato tramite il filtro di Kalman, un'altro modello e caso di studio sarà l'utilizzo di torrette Wi-fi integrate all'interno dell'ambiente che si vuole esplorare, in questo caso sarà invece implementato il particle filter. Una breve ricapitolazione del filtro di kalman sarà presentata mentre per il particle filter un'analisi più approfondita sarà condotta in quanto non argomento del corso.

iii.1 Filtro di Kalman

Un filtro di Kalman è uno stimatore ottimale, ovvero deduce i parametri di interesse da osservazioni indirette, imprecise e incerte. È ricorsivo, così nuove misure possono essere elaborate al loro arrivo. Se il rumore è gaussiano, il filtro di Kalman minimizza l'errore quadratico medio dei parametri stimati. Dato solo la media e la deviazione standard del rumore, il filtro di Kalman è il miglior stimatore lineare. Gli stimatori non lineari possono essere migliori.

iii.2 Filtro Particellare

L'algoritmo di localizzazione PF procede come segue: si inizializzano n particelle in una mappa. Ogni particella è un vettore di stato 3 per

1 del veicolo ad ognuna di queste si applica il modello di plant e se ne aggiunge un rumore al vettore di controllo u . Successivamente per ogni particella se ne prevede l'osservazione e si confronta questo con il valore misurato, tale confronto porterà al calcolo dell'innovazione o di ciò che definiremo peso della particella. Si selezionano le particelle che meglio spiegano l'osservazione, un modo per farlo è quello di costruire una pdf che descriva i campioni e i loro pesi, e poi rileszionare un nuovo set di particelle da questa pdf. La stima della posizione del robot fornita dal filtro e la media di questo nuovo ricampionamento

Il punto cruciale è che non richiede alcuna ipotesi di linearizzazione (non ci sono jacobiani coinvolti) e non ci sono ipotesi Gaussiane. È particolarmente adatto ai problemi con piccoli spazi di stato mentre in caso di vettori di stato grandi diventa computazionalmente pesante

V. IMPLEMENTAZIONE

Per la realizzazione del progetto si è optato per un simulazione in ambiente MATLAB, si è così realizzato un ambiente virtuale 2D atto a ricostruire la pianta di un edificio come, ad esempio, uffici, capannoni, aule. Allo scopo è stata usata una generazione procedurale. Il vantaggio più evidente dei livelli generati proceduralmente è la loro varietà che portano ad ogni esecuzione, l'ambiente cambia. Ciò significa che i robot non possono apprendere le posizioni di oggetti e questo permette di testare l'affidabilità in casi sempre differenti. Un altro vantaggio comune a tutte le implementazioni della generazione procedurale è il tempo che si risparmia nello sviluppo. Nel nostro progetto, avremo un numero infinito di livelli unici. Se stessimo creando i nostri livelli manualmente, ciò sarebbe semplicemente impossibile. Saremmo limitati a forse una decina di livelli al massimo. L'utilizzo di una generazione procedurale come questa elimina questo carico di lavoro dagli sviluppatori, risparmiando tempo e aumenta la portata di ciò che è possibile. Tra gli svantaggi, si ricorda, che la generazione procedurale di per sé non è in alcun modo casuale. L'assenza di control-

lo è una mancanza comune della generazione procedurale in generale, ma forse non è più prevalente rispetto alla generazione di ambienti. Dato che, di solito gli ambienti sono realizzati a mano da designer. Lasciare questo lavoro a un algoritmo si traduce in una significativa perdita di controllo. Un'altra considerazione che deve essere presa in considerazione è la potenza di calcolo che è richiesta, nel nostro caso, si ha solo una matrice 2D di piccole dimensioni che deve essere generata. Tuttavia, mappe di più larga scala, avranno un costo computazionale diventa più significativo e deve essere preso in considerazione.[6]

i. Binary Space Partitioning

Il partizionamento dello spazio binario è un processo generico di divisione ricorsiva di una scena in due finché il partizionamento soddisfa uno o più requisiti. Può essere visto come una generalizzazione di altre strutture ad albero spaziale, uno in cui gli iperpiani che suddividono lo spazio possono avere qualsiasi orientamento, piuttosto che essere allineati con gli assi delle coordinate.[7] L'albero ***k-d*** è un albero binario in cui ogni nodo è un punto k -dimensionale. Ogni nodo non foglia può essere pensato come generare implicitamente un iperpiano scisso che divide lo spazio in due parti, note come semispazi. I punti a sinistra di questo iperpiano sono rappresentati dal sotto-albero sinistro di quel nodo e i punti a destra dell'iperpiano sono rappresentati dal sotto-albero destro. La direzione dell'iperpiano viene scelta nel modo seguente: ogni nodo dell'albero è associato a una delle dimensioni k , con l'iperpiano perpendicolare all'asse di quella dimensione.[8]

ii. Esecuzione

Una volta generato lo scenario la simulazione inizia con il posizionamento di uno o più robot all'interno della mappa come condizione iniziale e assegnato un obiettivo da raggiungere. Questi non conoscono l'ambiente in cui si trovano e quindi tentano di raggiungere l'obiettivo nella maniera più rapida impostando un' traiettoria rettilinea determinata come norma

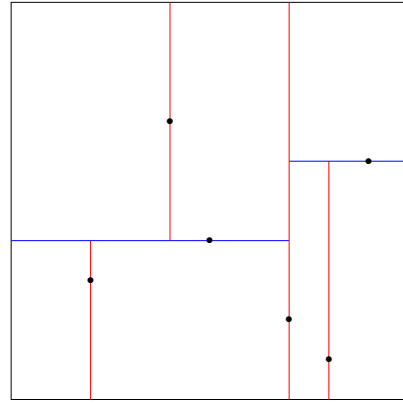


Figura 4: decomposizione per il set di punti.

tra due punti. La pianificazione della traiettoria viene modificata in base alle rilevazioni effettuate dal lidar in tal modo essi evitano l'ostacolo perché repulsi.

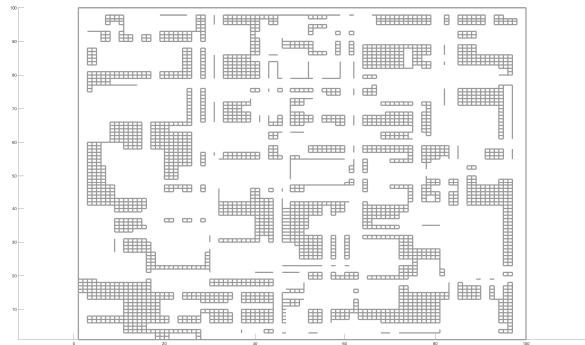


Figura 5: esempio di scenario procedurale 100×100

VI. RISULTATI

VII. CONCLUSIONI

E stato qui presentato un modo efficiente di condurre un'esplorazione coordinata di un ambiente. Possibile passo in avanti per la suddetta applicazione sarebbe la sua implementazione e simulazione fisica in modo da condurre uno studio più approfondito sulle possibili problematiche riscontrate. Migliorie software che possono essere condotte sono sicuramente una migliore rappresentazione fisica della comunicazione robot-robot robot-ancora wi fi tramite gli ultimi protocolli esistenti, lo studio più approfondito su un tipo di stima distribuita delle posizioni dei robot invece che individuale come attualmente risulta adottando algoritmi di least-square o di kalman distribuito. Altro limite attuale dell'applicazione è il non riconoscimento attuale di zone inaccessibili, ciò comporta nel caso di zone inaccessibili molto estese una gran probabilità di assegnazione di un target appartenente a tale zona per il robot.

- [1] B. Siciliano, *Robotica. Modellistica, pianificazione e controllo*, ser. Collana di istruzione scientifica. Serie di automatica. McGraw-Hill Companies, 2008. [Online]. Available: <https://books.google.it/books?id=HdaVPQAACAAJ>
- [2] C. K. M. C. K. A. N. H. Sabudin E. N, Omar. R, “Potential field methods and their inherent approaches for path planning,” *ARPJ Journal of Engineering and Applied Sciences*, vol. 11, no. 18, SEPTEMBER 2016.
- [3] X. Xu, C. Li, and Y. Zhao, “Air traffic rerouting planning based on the improved artificial potential field model,” in *2010 Chinese Control and Decision Conference*, May 2010, pp. 1444–1449.
- [4] Y. Koren and J. Borenstein, “Potential field methods and their inherent limitations for mobile robot navigation,” in *Proceedings. 1991 IEEE International Conference on Robotics and Automation*, Apr 1991, pp. 1398–1404 vol.2.
- [5] G. Li, A. Yamashita, H. Asama, and Y. Tamura, “An efficient improved artificial potential field based regression search method for robot path planning,” in *2012 IEEE International Conference on Mechatronics and Automation*, Aug 2012, pp. 1227–1232.
- [6] D. Green, *Procedural Content Generation for C++ Game Development*. Packt Publishing, 2016. [Online]. Available: <https://books.google.it/books?id=EQYcDAAAQBAJ>
- [7] W. contributors, “Binary space partitioning — wikipedia, the free encyclopedia,” 2018, [Online; accessed 14-March-2018]. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Binary_space_partitioning&oldid=828843135
- [8] —, “K-d tree — wikipedia, the free encyclopedia,” 2018, [Online; accessed 14-March-2018]. [Online]. Available: https://en.wikipedia.org/w/index.php?title=K-d_tree&oldid=829846868