

Progettazione di un sistema reattivo per la navigazione autonoma di un drone in ambienti sconosciuti

Luca Di Stefano

Relatore: Prof. Eliseo Clementini

Correlatore: Dott. Enrico Stagnini

Università degli Studi dell'Aquila

Dipartimento di Ingegneria e Scienze dell'Informazione e Matematica



UAV (Unmanned Aerial Vehicles)

Applicazioni in ambito civile:

- ▶ Monitoraggio, telerilevamento, raccolta dati
- ▶ Supporto alla produzione (ad es. agricoltura di precisione)
- ▶ Trasporto di pacchi e carichi postali. . .

UAS (Unmanned Aerial System)

- ▶ UAV
- ▶ GCS (*Ground Control Station* o “stazione di terra”)
- ▶ Sistemi di comunicazione



UAV Multirottore

- ▶ Costi ridotti
- ▶ *Payload* adeguati a molte applicazioni
- ▶ Elevata manovrabilità



Navigazione e guida autonoma

Missione: sequenza di operazioni da eseguire. Esempio:

1. Decolla; raggiungi 50 m di altezza dal suolo
2. Raggiungi il *waypoint* W_1
3. Acquisisci una fotografia
4. Ritorna al punto di decollo
5. Atterra



Navigazione e guida autonoma

Missione: sequenza di operazioni da eseguire. Esempio:

1. Decolla; raggiungi 50 m di altezza dal suolo
2. Raggiungi il *waypoint* W_1
3. Acquisisci una fotografia
4. Ritorna al punto di decollo
5. Atterra

Navigazione (*navigation*): attraversamento di uno o più *waypoint*

Guida (*guidance*): decisioni di alto livello relative alla missione: valutazione dell'andamento, modifica, cancellazione

Autonomia di un UAS: capacità di eseguire una missione senza l'intervento di un operatore umano o di un sistema esterno



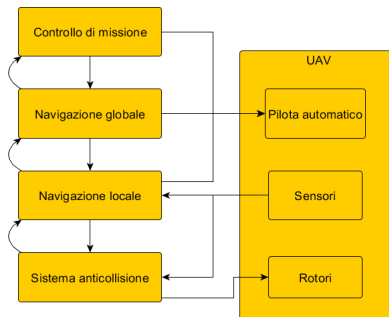
Stato dell'arte

Gerarchia di sottosistemi con obiettivi differenti

- ▶ Sistemi anticollisione (sensori di bordo)
Evitare l'impatto con altri oggetti
- ▶ Navigazione *locale* (sensori di bordo)
Raggiungere il prossimo *waypoint*
- ▶ Navigazione *globale* (mappe dell'area di missione)
Gestire il piano di volo (sequenza di waypoint)
- ▶ Controllo di missione (sistemi semantici/decisionali)
Modificare/annullare la missione; monitorarne l'andamento



Esempio di architettura stratificata



Pattern architetturale *Layers* (grado di astrazione crescente)
Sistema distribuito tra UAV e GCS

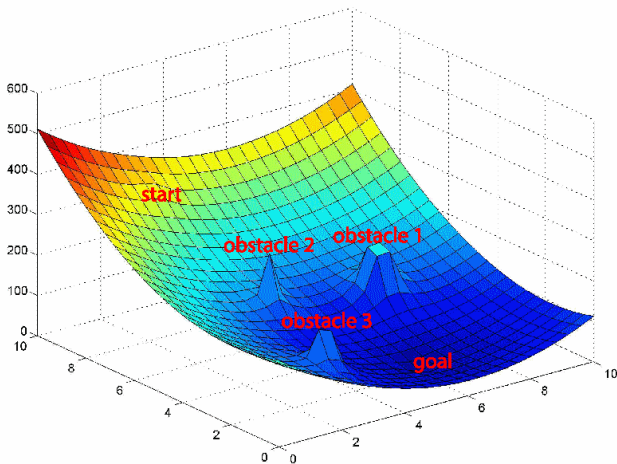


Navigazione locale

1. Costruzione di una mappa (grafo o DB di *voxels*) a partire dai dati dei sensori
2. Applicazione di uno dei seguenti metodi:
 - ▶ Cammino minimo su un grafo
 - ▶ Ottimizzazione numerica
 - ▶ Campi di potenziale



Campi di potenziale



Obiettivi

Sistema di navigazione locale autonoma

Requisiti

Capacità di operare a partire da pochi dati sull'ambiente

Elevata compatibilità

Basso costo computazionale

Architettura

Sistema anticollisione onboard (misura di sicurezza rispetto a folate di vento, malfunzionamenti ecc.)

Sistema di navigazione locale sulla GCS

- ▶ Mapless: non mantiene un modello dell'ambiente
- ▶ Reattivo: decisione in funzione degli stimoli sensoriali



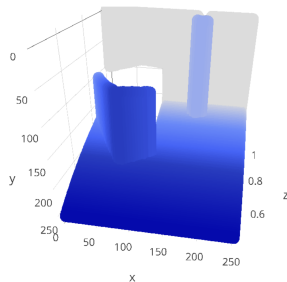
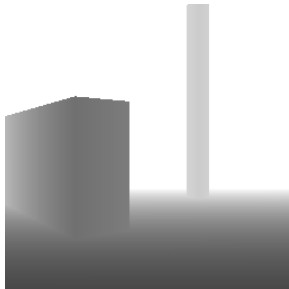
Mappa di profondità (depth map)

Acquisita dal drone: LIDAR, TOF, ecc.

Raster con pixel a valori *float* compresi tra 0 e 1

Valore = distanza normalizzata rispetto al *range* del sensore

L'angolo di visione (orizzontale/verticale) del sensore è noto:
possiamo risalire alla posizione 3D dell'ostacolo rispetto al drone



Visibilità

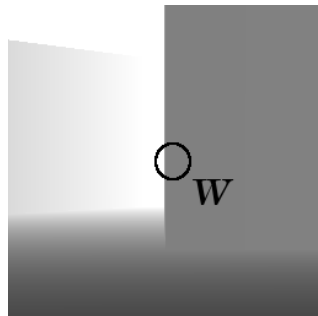
Note le coordinate (GPS) di W e del drone, possiamo:

1. Ottenere le coordinate sferiche di W rispetto al drone
2. Proiettare W sulla mappa di profondità (pixel W')
3. Recuperare i valori della depth map in un intorno di W'

Confrontando i valori con la distanza da W possiamo determinare se un ostacolo ostruisce la visibilità di W .

Il raggio dell'intorno dipende da:

- ▶ Raggio del drone
- ▶ Distanza drone-waypoint
- ▶ Un margine di sicurezza (configurabile)



Descrizione generale dell'algoritmo

Input: un waypoint obiettivo T

1. $W = T$
2. Finché il drone non si trova in T :
 - 2.0 Se il drone si trova in W e $W \neq T$:
 - $W = T$
 - 2.1. Orienta il drone verso W
 - 2.2. Acquisisci la depth map
 - 2.3. Valuta se W è visibile:
 - Se è visibile, raggiungilo
 - Altrimenti:
 - Se $W == T$: $W = (\text{waypoint intermedio})$
 - Altrimenti $W = T$



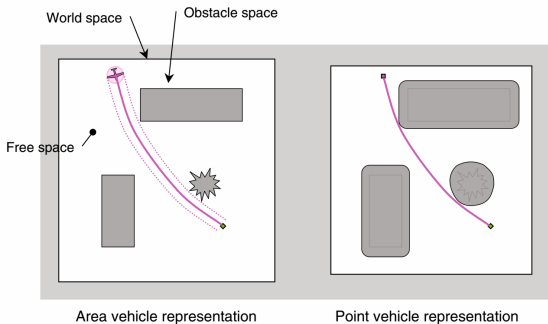
Replanning (1)

Se l'obiettivo non è visibile, bisogna generare un nuovo waypoint

W_{new} nello spazio visibile. $DW_{new} \leq DW$

Dobbiamo tener conto del volume del drone!

In 2D si ricorre alla *dilatazione* della mappa degli ostacoli:



Replanning (2)

Dividiamo la depth map in n strati in base ai valori dei pixel
Gli strati più vicini al drone vengono dilatati maggiormente

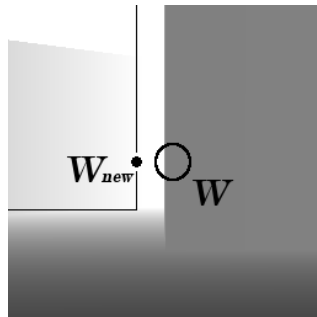


Replanning (3)

Impostiamo a 1 tutti i pixel superiori a un valore di soglia
Scegliamo il migliore tra i pixel adiacenti all'ostacolo dilatato

Otteniamo così ϑ , φ del nuovo waypoint.

ρ = valore del primo pixel < 1
sul segmento che va da W_{new} a
 W (bordo dell'ostacolo)



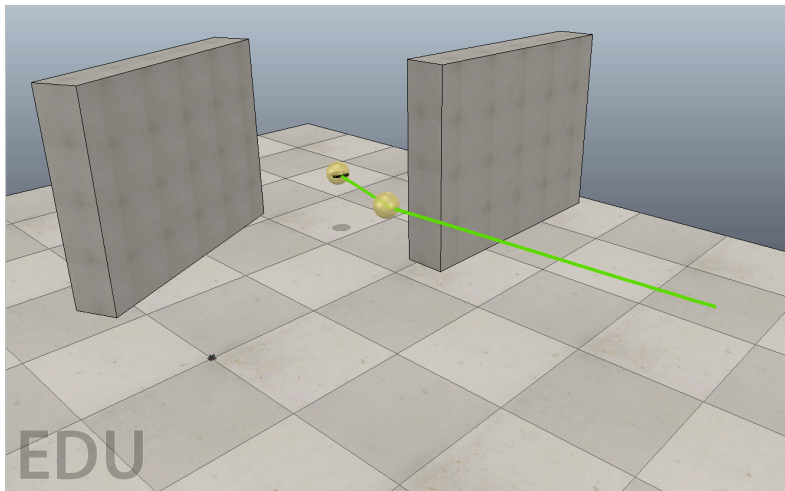
Piattaforma di simulazione

- ▶ Framework di simulazione V-REP
- ▶ Algoritmo scritto in Python 3.5
 - ▶ API V-REP
 - ▶ Numpy
 - ▶ OpenCV

Grazie a classi *adapter* che gestiscono la comunicazione con V-REP si prevede una rapida portabilità su un'implementazione reale



Scenario 1



Scenario 2 - Descrizione



12 simulazioni

Risoluzione del LIDAR: da 256x256 a 32x32

AOV fisso, 90° (orizzontale/verticale)

Bassa velocità del drone (0.6 m/s)

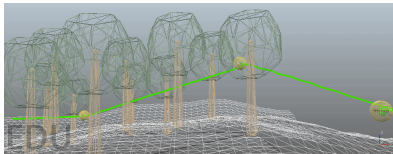


Scenario 2 - Risultati (1)

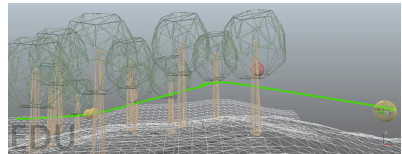
#	Risoluzione	Waypoint		Distanza (m)		Tempo
		Raggiunti	Scartati	Assoluta (m)	Normalizzata	
1	256	2	0	24.16	1.04	00:58.6
2	256	2	1	23.83	1.03	01:12.0
3	256	3	0	23.43	1.01	01:22.5
4	128	2	1	23.62	1.02	01:13.0
5	128	4	1	24.08	1.03	01:32.0
6	128	3	1	23.54	1.01	01:12.1
7	64	4	5	24.99	1.07	02:09.6
8	64	3	2	24.19	1.04	01:20.7
9	64	3	1	23.68	1.02	01:20.3
10	32	4	1	23.73	1.02	01:25.8
11	32	3	2	24.05	1.03	01:46.2
12	32	3	2	23.79	1.02	01:18.3



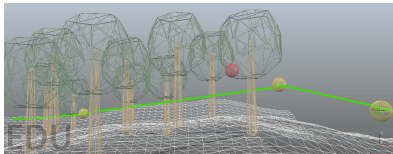
Scenario 2 - Risultati (2)



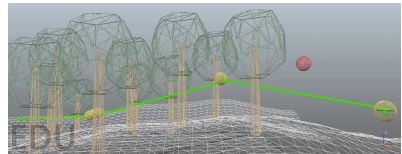
(a) Simulazione 1.



(b) Simulazione 4.



(c) Simulazione 9.



(d) Simulazione 10.



Conclusioni

- ▶ Buona robustezza rispetto alla risoluzione della depth map
- ▶ Alta qualità dei risultati rispetto alla distanza ottima

Possibili sviluppi futuri

- ▶ Implementazione reale
- ▶ Adattamento dinamico dei margini di sicurezza in base alla qualità del *fix* GPS
- ▶ Elaborazione della depth map per ricostruire una mappa degli ostacoli (supporto a successive missioni nella stessa area)

