

ALMA MATER STUDIORUM - UNIVERSITÀ DI BOLOGNA
CAMPUS DI CESENA

DIPARTIMENTO DI INGEGNERIA DELL'ENERGIA
ELETTRICA E DELL'INFORMAZIONE
"GUGLIELMO MARCONI"

Corso di Laurea in Ingegneria Elettronica per l'Energia e
l'Informazione

Radar perimetrali per applicazioni automotive

Elaborato in
Comunicazioni Digitali e Internet

Relatore

Chiar.mo Prof. Ing.
DAVIDE DARDARI

Candidata

DANIA DE CRESCENZO

Correlatori

Prof. Ing.
GIANNI PASOLINI
Dott. Ing.
ANNA GUERRA

Anno Accademico 2020 – 2021

Indice

Introduzione	1
1. Il sistema RADAR	3
1.1 Principi fondamentali della tecnologia radar.....	4
1.2 Il radar FMCW	6
2. Mapping mediante segnali radar	14
2.1 Mappe a griglia di occupazione.....	15
2.2 Mappe a griglia d'ampiezza	20
2.3 Mapping collaborativo	22
3. Radar Infineon BGT60ATR24C	24
3.1 Parametri di controllo.....	25
3.2 Configurazione	28
4. Campagna sperimentale	31
4.1 Esperimenti preliminari	31
4.2 Esperimenti in scenario reale	37
5. Elaborazione dati e risultati	45
Conclusioni	62
A. Sensing GUI	65
A.1 Configurazione del frame	65
A.2 Acquisizione dati	67
B. Codice Matlab	71
Lista delle figure	79
Bibliografia	83

Introduzione

Attualmente, con il rapido sviluppo dei veicoli automatizzati, una delle principali tendenze tecnologiche riguarda la ricerca nel campo della percezione ambientale. L'utilizzo di sensori di rilevazione svolge infatti un ruolo importante nel miglioramento delle prestazioni del veicolo: grazie alle ultime tecnologie è possibile ottenere una guida sicura, confortevole ed efficiente utilizzando una combinazione di diversi tipi di dispositivi in grado di implementare funzioni come l'auto-localizzazione ad alta precisione, il processo decisionale, e la pianificazione del movimento.

Alcuni strumenti ampiamente utilizzati a questo scopo sono i sensori visivi, che forniscono informazioni semantiche ad alta precisione, oppure i LIDAR (*Laser Imaging Detection and Ranging*), utilizzati per le loro ottime prestazioni sulla lunga distanza e sull'ampiezza del campo visivo.

Tra gli altri trova spazio la tecnologia RADAR (*Radio Detection and Ranging*), largamente sfruttata per il suo basso costo e per la sua robustezza rispetto alle variazioni di luminosità e condizioni meteo. Nelle auto moderne i sensori radar sono ormai parte dell'equipaggiamento standard, utilizzati ad esempio per la regolazione automatica della velocità e la prevenzione di collisioni. In particolare, il radar a onda continua modulata in frequenza (*frequency modulated continuous wave radar – FMCW*) è un dispositivo che trasmette, e di conseguenza riceve, onde continue, tipicamente caratterizzate da un andamento sinusoidale e di frequenza crescente nel tempo (chiamate *chirp*). Esso è ad oggi un elemento percettivo comune e necessario sui veicoli automatizzati, e consente una lunga distanza di misurazione, capacità di rilevamento dinamico del bersaglio e adattabilità ambientale; caratteristiche che migliorano la stabilità, la sicurezza e l'affidabilità complessive del veicolo rilevando e reagendo agli ambienti esterni e interni.

Il lavoro di tesi consiste in uno studio sperimentale inerente a questo tipo di sensore, trattando nello specifico il modello di radar FMCW BGT60ATR24C di Infineon, pensato per applicazioni di tipo automobilistico a corto raggio.

L'obiettivo finale è quello di eseguire il mapping tramite segnali radar dell'ambiente statico circostante il veicolo, al fine di supportare l'implementazione di applicazioni volte a identificare ostacoli potenzialmente pericolosi durante l'apertura dello sportello. L'attività di tesi è organizzata come segue.

Il capitolo 1 presenta una breve introduzione ai principi della tecnologia radar, ponendo particolare attenzione sui dispositivi FMCW e le informazioni d'ambiente che questi permettono di rilevare.

Nel capitolo 2 si esamina il problema del mapping tramite segnali radar, valutando diversi approcci alla costruzione della mappa e accennando al mapping collaborativo, tecnica che consiste nella fusione di dati provenienti da più sensori. Ci si concentra soprattutto sul mapping probabilistico a griglia di occupazione, tecnica che verrà poi applicata ai dati ottenuti sperimentalmente.

Nel capitolo 3 si fornisce una descrizione generale del radar BGT60ATR24C a disposizione, si illustrano i parametri di controllo e la configurazione scelta per eseguire le acquisizioni sul campo.

Il capitolo 4 riguarda la campagna sperimentale, divisa in esperimenti preliminari in ambiente controllato ed esperimenti in uno scenario statico automobilistico reale.

Nel capitolo 5 verrà infine illustrata l'elaborazione Matlab dei dati sperimentali raccolti, arrivando alla produzione di una mappa 3D che rappresenta probabilisticamente lo spazio osservato da due radar posizionati sulla fiancata di un'automobile. Le misurazioni effettuate dai due sensori sul medesimo scenario statico verranno quindi fuse in un'unica rappresentazione dell'ambiente, in modo da aumentare l'accuratezza e l'affidabilità della mappa rispetto al caso in cui si utilizzi un unico dispositivo di rilevamento.

1 Il sistema RADAR

Il RADAR, dall'inglese "*RADio Detection And Ranging*", è un sistema elettronico in grado di individuare la presenza di oggetti nello spazio circostante, detti comunemente target, utilizzando onde elettromagnetiche a radiofrequenza [1].

Il suo funzionamento si basa su un fenomeno fisico chiamato "*backscattering*", che consiste nella dispersione della radiazione elettromagnetica a seguito dell'interazione tra questa e un oggetto di dimensioni maggiori della lunghezza d'onda della radiazione incidente.

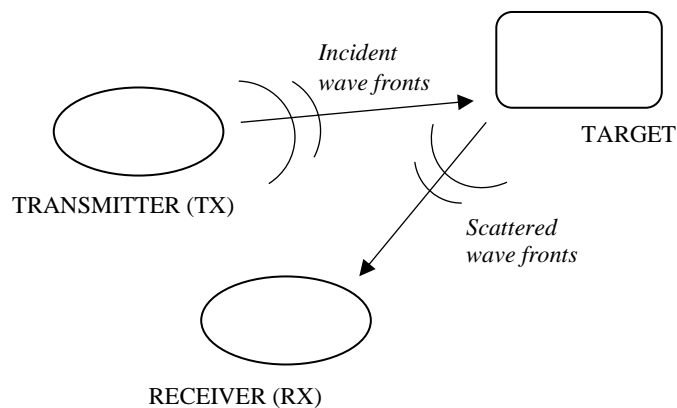


Figura 1.1 Schema a blocchi di un generico sistema radar

Il sistema Radar nella sua configurazione più semplice è composto due elementi: un trasmettitore ed un ricevitore, come mostrato in figura 1.1. Il trasmettitore emette una forma d'onda che investe l'ostacolo e ritorna sottoforma di "eco" al ricevitore, che elaborandola ricava informazioni riguardanti il target, ad esempio, circa la sua distanza, velocità radiale, l'angolo di vista e dimensione, dipendentemente dal tipo di radar. In particolare, la radiazione di ritorno può essere rilevata dall'antenna ricevente con un certo ritardo, tipicamente pari al doppio del tempo di propagazione antenna-target; conoscendo la velocità di propagazione dell'onda elettromagnetica nel mezzo considerato è quindi possibile risalire alla

distanza del bersaglio e alla sua posizione angolare rispetto al sistema di riferimento se si usano più antenne (radar multiple-input multiple-output - MIMO). Utilizzando segnali trasmessi modulati in frequenza è inoltre possibile stimare la velocità di target in movimento, sfruttando il principio dell'effetto Doppler.

1.1 Principi fondamentali della tecnologia radar

In generale, in un sistema radar, il segnale generato da un trasmettitore è irradiato in atmosfera da un'antenna trasmittente (TX). L'onda elettromagnetica, dopo essersi propagata, viene riflessa dal target ed è processata dal ricevitore (RX). Il segnale ricevuto subisce diverse interferenze durante il suo percorso, quali il rumore dei componenti elettronici e il rumore ambientale, l'interferenza elettromagnetica prodotta da altri segnali esterni al sistema, l'interferenza dovuta a riflessioni su altri target non d'interesse (clutter), e i disturbi dovuti ad altri dispositivi elettronici nelle vicinanze. Lo scopo del ricevitore è quindi quello di individuare il segnale tra i disturbi, per poi elaborarlo.

I sistemi radar si possono classificare in base al tipo di onda emessa dal trasmettitore: i *Continuous Wave Radar* trasmettono continuamente segnale nel mezzo, mentre i *Pulsed Wave Radar* trasmettono impulsi finiti di breve durata, per ovviare al problema della sovrapposizione tra segnale trasmesso e ricevuto. È possibile operare una distinzione anche in base alla configurazione del TX e del RX. Tipicamente, il TX e RX sono co-locali e utilizzano la medesima antenna sia per la trasmissione sia per la ricezione, per cui si parla di *radar mono-statico*. Nel caso in cui il TX e RX sono separati si ottiene un radar *bi-statico* o *multi-statico* se sono presenti più RX e TX. I radar multi-statici sono particolarmente vantaggiosi in quanto migliorano la capacità di rilevamento (detection) di un target grazie all'osservazione da più punti di vista.

Con riferimento al caso classico di radar mono-statico, l'equazione fondamentale del radar [2], ovvero dal bilancio del

radiocollegamento applicato alla tecnologia radar, che mette in relazione la potenza ricevuta con la potenza trasmessa e la distanza radar-target.

Sia P_T la potenza trasmessa e G_T il guadagno dell'antenna trasmittente, la densità di potenza a distanza R sarà:

$$S_T = \frac{P_T G_T}{4\pi R^2} \quad [W/m^2] \quad (1.1)$$

La potenza incidente un target a distanza R decresce quindi con il quadrato della distanza tra radar e target. La densità di potenza ricevuta dal radar a fronte di una riflessione sul target si esprime invece come:

$$S_R = \frac{P_T G_T}{4\pi R^2} \frac{\sigma}{4\pi R^2} \quad [W/m^2] \quad (1.2)$$

dove σ è detta “*Radar Cross Section*” (RCS), parametro che misura la capacità del target di riflettere l'onda incidente. Infatti maggiore è l'RCS, maggiore è la possibilità che l'oggetto sia individuato correttamente dal radar. Si definisce area efficace A_R dell'antenna ricevente:

$$A_R = \frac{G_R \lambda^2}{4\pi} \quad [m^2] \quad (1.3)$$

con G_R guadagno dell'antenna ricevente e λ lunghezza d'onda del segnale. Per ottenere invece la potenza ricevuta P_R occorre moltiplicare la densità di potenza ricevuta S_R per l'area efficace A_R dell'antenna ricevente ottenendo così l'equazione fondamentale del radar:

$$P_R = \frac{P_T G_T \sigma G_R \lambda^2}{(4\pi)^3 R^4} \quad [W] \quad (1.4)$$

Utilizzando questa equazione è possibile ricavare parametri come P_T o λ avendo noto il valore desiderato di P_R minima, in base alla sensibilità del ricevitore. È importante sottolineare che la potenza ricevuta è inversamente proporzionale alla quarta potenza della distanza dall'ostacolo e direttamente proporzionale al quadrato della lunghezza d'onda del segnale.

1.2 Il radar FMCW

Tra i radar a onda continua, molto comuni in applicazioni industriali e nel campo automotive, si trovano i “*Radar ad onda continua modulata in frequenza*” (frequency modulation continuous wave -FMCW Radar) [1]. Questi sistemi sono in grado di individuare un target e stimare diversi parametri quali velocità e distanza del target.

La tecnica FMCW è basata sul “chirp”, un segnale sinusoidale la cui frequenza aumenta linearmente nel tempo da un valore minimo ad un valore massimo. Il segnale è caratterizzato da una larghezza di banda B , una frequenza iniziale f_c e una durata T_c . Un modo agevole di rappresentare graficamente il chirp è quello mostrato in figura 1.2, dove il grafico frequenza-tempo raffigura la retta di pendenza S , corrispondente alla velocità di aumento della frequenza del chirp e pari al rapporto tra banda e durata del segnale.

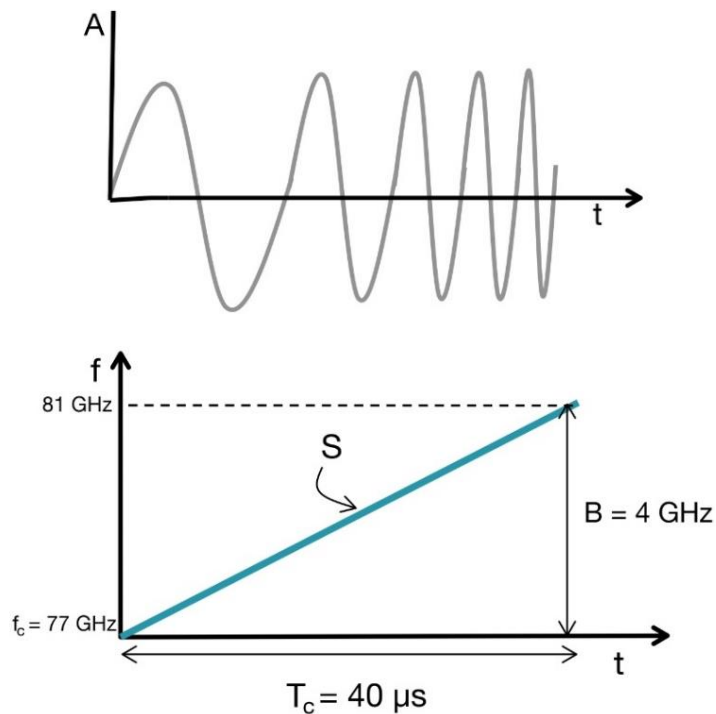


Figura 1.2 Segnale 'chirp' nel tempo

In figura 1.3 è rappresentato il diagramma a blocchi di un semplice sistema radar FMCW [2]. Il sintetizzatore genera il chirp, definito per t in $[0, T_C]$, che viene trasmesso dall'antenna trasmittente:

$$s_T(t) = A \sin(2\pi f_{TX}t) \quad (1.5)$$

dove la frequenza dipende dal tempo secondo la relazione:

$$f_{TX}(t) = f_c + St \quad (1.6)$$

Nell'ipotesi di scenario ideale in spazio libero con un solo target posto a distanza R , il chirp viene quindi riflesso dal target e ricevuto dall'antenna RX:

$$s_R(t) = B \sin(2\pi f_{RX}(t - \tau)) \quad (1.7)$$

dove la frequenza si esprime:

$$f_{RX}(t) = f_c + S(t - \tau) \quad (1.8)$$

Il circuito elettronico chiamato mixer moltiplica i segnali TX ed RX generando così un segnale con frequenza pari alla differenza tra le frequenze dei due ingressi e fase pari alla differenza tra le fasi dei due ingressi, ed un segnale somma che viene poi filtrato dal filtro IF.

Il segnale prodotto dal mixer è detto "segnale IF" a frequenza intermedia $f_{IF} = f_{TX} - f_{RX}$ e permette di calcolare la distanza dal target. Il chirp captato dall'antenna ricevente ha un ritardo rispetto al chirp trasmesso definito come "round trip time" $\tau = 2R/c$, dove R è la distanza dal target e c la velocità della luce nel mezzo.

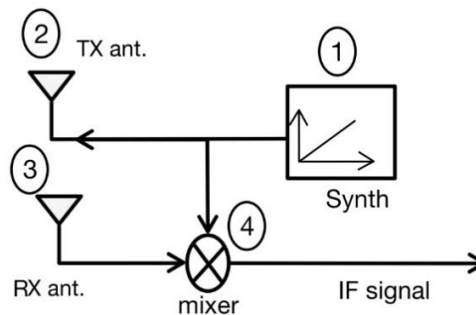


Figura 1.3 Schema a blocchi di un sistema radar FMCW

La figura 1.4 mostra il chirp trasmesso e il chirp ricevuto (entrambi con pendenza S); si nota chiaramente che il segnale ricevuto è una versione del segnale trasmesso ritardata di un tempo τ .

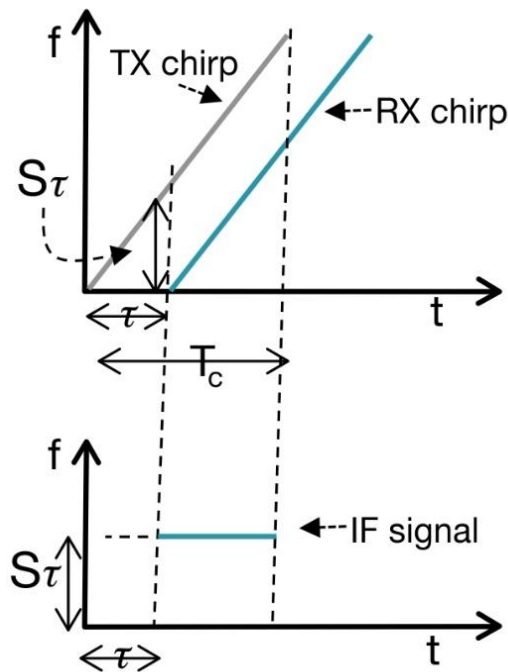


Figura 1.4 TX chirp e RX chirp

Se si considera un unico target nel campo visivo del radar, per ottenere la rappresentazione della frequenza del segnale IF in funzione tempo è necessario sottrarre le due linee presentate nella sezione superiore della figura 1.4. La distanza tra le due rette è fissa, il che significa che il segnale IF è costituito da un tono sinusoidale con frequenza costante:

$$s_{IF}(t) = C \sin(2\pi f_{IF}t + \varphi_0) \quad (1.9)$$

$$f_{IF} = S\tau = \frac{S2R}{c} \quad (1.10)$$

è importante sottolineare che il segnale IF è valido solo nell'intervallo di tempo in cui sia il chirp TX che il chirp RX si sovrappongono (ovvero, l'intervallo tra le linee tratteggiate verticali in figura 1.4).

La fase del segnale IF è la differenza tra la fase del chirp TX e la fase del chirp RX nell'istante corrispondente all'inizio del segnale IF (cioè l'istante temporale rappresentato dalla verticale sinistra linea tratteggiata in figura 1.4):

$$\varphi_0 = 2\pi f_{IF}\tau = \frac{4\pi R}{\lambda} \quad (1.11)$$

Quindi misurando f_{IF} si può ricavare la distanza tra radar e target invertendo l'equazione (1.10):

$$R = \frac{f_{IF}c}{2S} \quad (1.12)$$

D'altro canto, se sono presenti diversi oggetti, il segnale ricevuto sarà composto da diversi toni sinusoidali. In questo caso si utilizza la trasformata di Fourier per separare i toni nel dominio delle frequenze. È noto che una senoide nel tempo produce un singolo picco nelle frequenze; in presenza di più toni sinusoidali sovrapposti (ovvero più target nell'ambiente) generalmente una finestra di osservazione temporale T_C è sufficiente per distinguere picchi in frequenza separati da $\Delta f > 1/T_C$ Hz. Se le frequenze delle due sinusoidi sono troppo vicine rispetto alla T_C considerata, esse verranno visualizzate come un singolo picco. Per due oggetti separati di Δd in direzione radiale, la differenza tra le loro frequenze IF è data da $\Delta f = 2S\Delta d/c$, inoltre per distinguerli deve essere che $\Delta f > 1/T_C$ allora $2S\Delta d/c > 1/T_C$ e quindi $\Delta d > c/2B$ con $B = ST_C$.

L'abilità del radar di distinguere diversi oggetti distanti Δd è detta "*range resolution*", definita come distanza minima tra due oggetti distinguibili:

$$d_{res} = \frac{c}{2B} \quad (1.13)$$

Per aumentare la risoluzione spaziale è necessario aumentare la durata della finestra di osservazione T_C del chirp finché i toni sinusoidali non sono sufficientemente separati in frequenza; questo

porta ad un aumento di banda. Si nota che il range di risoluzione dipende solo dalla larghezza di banda spazzata dal chirp: più è larga B maggiore è la range resolution e minore è la distanza tra i target che il radar riesce a discriminare.

Come già menzionato in precedenza, i radar FMCW permettono di misurare anche la velocità del target in movimento. Nel caso di singolo target, vengono trasmessi due chirp dall'antenna TX, separati di un tempo pari a T_C , quindi al ricevitore verranno captate due sinusoidi, entrambe a frequenza IF ma con fasi diverse. La fast Fourier transform (FFT) applicata ad ognuno dei due chirp avrà picco nella stessa posizione, ma con fase diversa. La differenza di fase misurata corrisponde a uno spostamento dell'oggetto pari a:

$$\Delta d = vT_C \quad (1.14)$$

La differenza di fase, derivata dall'equazione (1.11), si esprime come:

$$\Delta\varphi = \frac{4\pi\Delta d}{\lambda} \quad (1.15)$$

infine, sostituendola (1.14) ed invertendo la (1.15) si ottiene:

$$v = \frac{\lambda\Delta\varphi}{4\pi T_C} \quad (1.16)$$

La velocità dipende chiaramente dalla differenza di fase; per questo se $|\Delta\varphi| < \pi$ si ha ambiguità di misura dovuta alla periodicità di 2π della fase. Quindi, utilizzando la (1.16), si deriva la massima velocità che il radar può individuare utilizzando due chirp trasmessi a distanza T_C [3]:

$$v_{max} = \frac{\lambda}{4T_C} \quad (1.17)$$

Per ottenere una velocità massima misurabile maggiore è necessario diminuire il tempo trascorso tra la trasmissione dei due chirp.

La “*velocity resolution*” [3], definita come minima differenza di velocità percepibile tra due oggetti equidistanti dal radar, può essere

determinata trasmettendo una sequenza di N chirp equi-spaziati, detta frame, ed è definita come:

$$v_{res} = \frac{\lambda}{2T_s} \quad (1.18)$$

dove il tempo di frame è:

$$T_s = NT_C \quad (1.19)$$

Maggiore è il tempo di frame maggiore sarà la velocity resolution.

Inoltre un sistema radar è in grado di stimare l'angolo di arrivo (*Angle of Arrival, AoA*) del segnale ricevuto. Per farlo sono necessarie minimo due antenne riceventi (nel caso di un unico target), come mostrato in figura 1.5.

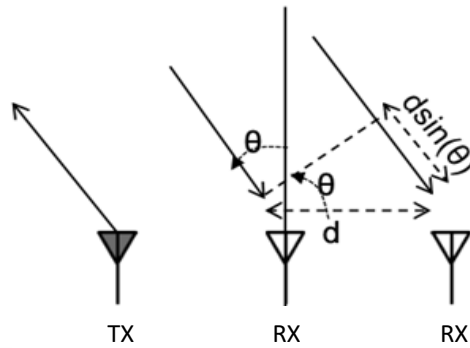


Figura 1.5 Angolo di arrivo del segnale ricevuto

Si considera la variazione di fase tra i segnali riflessi da un unico ostacolo e ricevuti dalle due antenne RX, legata alla distanza tra ogni antenna e il target. Questa si può esprimere come:

$$\Delta\varphi = \frac{2\pi\Delta d}{\lambda} \quad (1.20)$$

con $\Delta d = d \sin(\theta)$, e d distanza tra le due antenne riceventi. È possibile quindi stimare l'angolo di arrivo del segnale ricevuto:

$$\theta = \sin^{-1} \left(\frac{\lambda \Delta\varphi}{2\pi d} \right) \quad (1.21)$$

Si può osservare dalla (1.21) come la relazione tra θ e $\Delta\varphi$ è non lineare: a $\theta = 0$, $\Delta\varphi$ è quindi più sensibile alle variazioni dell'angolo di arrivo θ , mentre la sensibilità di $\Delta\varphi$ all'angolo θ si riduce quando θ tende a 90° . La stima dell'angolo di arrivo è infatti più accurata vicino allo 0.

Il massimo campo di visione che può essere individuato da due antenne spaziate di d è definito come:

$$\theta_{max} = \sin^{-1}\left(\frac{\lambda}{2d}\right) \quad (1.22)$$

Infine l'angolo di risoluzione, cioè il minimo angolo di separazione tra i target in modo che essi vengano distinti correttamente dal radar è definito come:

$$\theta_{res} = \frac{\lambda}{Nd\cos(\theta)} \quad (1.23)$$

Con N numero di antenne disponibili, fattore che aumenta la risoluzione. Si nota inoltre come θ_{res} dipenda dall'angolo di arrivo θ , e come migliori con θ tendente a 0 [3].

I sistemi di antenne multiple come quello appena considerato, sono oggetto di ricerca intensiva negli ultimi anni [4]. Il termine radar single-input-multiple-output (SIMO) si riferisce a un dispositivo radar con una singola trasmissione e più antenne di ricezione. La risoluzione angolare di un radar SIMO dipende dal numero di antenne RX. Ad esempio, un dispositivo con quattro antenne RX ha una risoluzione angolare di circa 30° , mentre un dispositivo con otto antenne RX ha una risoluzione angolare di circa 15° . Pertanto, un approccio diretto al miglioramento della risoluzione angolare richiede l'aumento del numero di antenne RX. Il termine multiple-input-multiple-output (MIMO) si riferisce a un sistema radar con più antenne TX e più antenne RX. Sviluppati inizialmente nella scienza della comunicazione per migliorare le funzionalità fondamentali quali la copertura e la qualità dei dati, i radar MIMO irradiano simultaneamente segnali non correlati in più direzioni o in una sola direzione, a seconda della configurazione. Questi sistemi sono caratterizzati da una migliore risoluzione angolare e possono inoltre fornire una robusta immunità alle interferenze; in generale l'architettura fornisce un insieme di Nt antenne

di trasmissione e N_r antenne di ricezione che possono visualizzare l'obiettivo da diverse angolazioni. Nel caso di più antenne TX, tipicamente queste si alternano in trasmissione mentre il segnale riflesso viene ricevuto contemporaneamente dalle antenne ad uso RX. Un sistema MIMO di questo tipo, con $N_t \times N_r$ antenne in TX e RX, è equivalente ad un sistema SIMO con una sola antenna TX e $N_t \times N_r$ antenne RX, dette comunemente antenne virtuali.

2 Mapping mediante segnali radar

Le rappresentazioni dell'ambiente tramite mappe a griglia sono un comune livello di astrazione in applicazioni di robotica e automotive. Sono infatti varie le funzioni basate su sensori radar che utilizzano mappe a griglia, ad esempio il rilevamento di spazio libero e di ostacoli, la stima, la classificazione e la localizzazione del percorso stradale [5].

Gli algoritmi di mapping “*grid-based*” discretizzano l'ambiente in uno schema 2D o 3D composto da celle, ed aggiornano il valore di ciascuna cella a ogni rilevamento. Pertanto, per associare le misurazioni alla giusta posizione nell'ambiente, cioè alla loro cella corrispondente sulla griglia, la posizione e l'orientamento del sensore durante l'acquisizione delle misurazioni devono essere note.

Mentre un veicolo è in marcia, i suoi sensori catturano continuamente dati sull'ambiente circostante; per ogni campione istantaneo raccolto il radar può ricavare la distanza, l'angolo di arrivo e la velocità radiale relativa degli oggetti presenti nel campo visivo.

L'integrazione delle misurazioni di più sensori radar in un'unica griglia è pratica comune ed è utile a diversi scopi: innanzitutto l'area totale della griglia non è costretta al singolo campo visivo di un sensore permettendo di avere una visione d'insieme; occlusioni momentanee dovute ad esempio ad interferenze causate da altri veicoli possono essere compensate da altre misurazioni; l'incertezza della misurazione del singolo frame acquisito dal radar può essere ridotta.

Diverse applicazioni possono richiedere diverso grado di precisione della griglia. Ad esempio applicazioni per la guida autonoma si concentrano sulla distinzione di ostacoli e spazi liberi, utilizzando quindi una mappa a griglia di occupazione in cui l'RCS del target è poco importante, dato che sia ostacoli “forti” (i.e. cartelli stradali) sia ostacoli “deboli” (i.e. vegetazione) devono essere ugualmente individuati ed evitati. Invece, se oltre alla posizione è necessario conoscere anche il tipo di target sfruttando l'RCS, si utilizzano le mappe a griglia d'ampiezza. In questo caso per ogni cella viene valutata anche l'ampiezza del segnale ricevuto dal radar, per ricavare le caratteristiche del target che ne permettono il riconoscimento.

2.1 Mappe a griglia di occupazione

Gli algoritmi di mapping a griglia di occupazione (*Occupation Grid-Map* – OGM) utilizzano una rappresentazione dell'ambiente che divide la mappa in celle di dimensione stabilita.

I segnali catturati dal radar vengono inizialmente processati al fine di ricavare le coordinate (x, y, z) del presunto target rispetto alla posizione del sensore, ottenendo così una nuvola di punti che rappresentano gli ostacoli rilevati. A questo punto il problema è quello di discriminare le rilevazioni dovute a target reali e quelle erronee, causate dal rumore. A tal proposito tipicamente si sottopone la nuvola di punti a un certo livello di soglia: i punti con intensità al di sopra della soglia sono ritenuti bersagli, mentre quelli al di sotto sono considerati rumore, e quindi eliminati.

Supponendo di partire da una nuvola di punti radar stabile e non rumorosa, l'aggiornamento della mappa a griglia di occupazione si riduce banalmente alla sovrapposizione di tutti i dati rilevati nel tempo. Tale supposizione è tuttavia poco plausibile in uno scenario reale, di conseguenza viene tipicamente impiegato un processo di fusione probabilistico in modo da filtrare rilevazioni erronee dovute a rumore o riflessioni, ottenendo così una OGM il più possibile aderente alla realtà.

Per semplicità di trattazione si considera nel seguito una OGM a due dimensioni (x, y) , che rappresenta uno scenario statico. La generica mappa a griglia di occupazione m in (x, y) esprime tramite una variabile binaria lo stato di ogni cella. Quando la cella i è occupata si ha:

$$m_i = 1 \quad (2.1)$$

altrimenti, quando la cella è libera:

$$m_i = 0 \quad (2.2)$$

con $i = 1, 2, 3, \dots, M$ per una mappa composta da M celle. Lo stato complessivo della mappa può essere rappresentato tramite il seguente vettore:

$$\mathbf{m} = [m_1, m_2, m_3, \dots, m_M] \quad (2.3)$$

Le informazioni riguardo la posizione dei target ottenute dalle misurazioni del sensore radar vengono utilizzate per stimare la probabilità a posteriori che ogni cella della mappa sia occupata da un ostacolo o meno:

$$P(\mathbf{m}|z_{1:t}, x_{1:t}) \quad (2.4)$$

dove \mathbf{m} è la mappa considerata, $z_{1:t}$ è l'insieme delle misure rilevate dal tempo 1 al tempo t , e $x_{1:t}$ è l'insieme delle posizioni del radar dal tempo 1 al tempo t (che si presumono note). Il costo computazionale legato alla stima a posteriori di $P(\mathbf{m}|z_{1:t}, x_{1:t})$ è dovuto alla dimensionalità del problema: se la mappa \mathbf{m} contenesse ad esempio 10.000 celle, allora il numero di mappe che potrebbero essere rappresentate dalla suddetta griglia è $2^{10.000}$. Pertanto il calcolo esaustivo di una probabilità a posteriori per tutte le possibili mappe presenterebbe molte difficoltà.

Sia m_i la cella della griglia con indice i , la notazione $P(m_i)$ denota la probabilità che la cella i sia occupata. L'approccio standard è quello di scomporre il problema totale in problemi più piccoli, stimando la probabilità:

$$P(m_i|z_{1:t}, x_{1:t}) \quad (2.5)$$

per tutte le celle m_i della griglia. Ciascuno di questi problemi di stima è quindi un problema binario. Questa suddivisione presuppone l'ipotesi che le celle non siano in alcun modo correlate tra loro. Assumendo quindi che le celle siano indipendenti è possibile ridurre la complessità del problema modellando la mappa come prodotto delle probabilità marginali delle singole celle:

$$P(\mathbf{m}|z_{1:t}, x_{1:t}) = \prod_i P(m_i | z_{1:t}, x_{1:t}) \quad (2.6)$$

Il classico algoritmo di mappatura a griglia di occupazione, trattato ampiamente in [6], [7] e [8], è basato su tre assunzioni principali: ambiente statico, celle indipendenti tra loro e posizione del sensore nota. Verificate queste ipotesi semplificative la costruzione della OGM si fonda essenzialmente su due elementi. Il primo è un modello a sensore inverso (*inverse sensor model* – ISM), un modello di probabilità di occupazione condizionale delle celle, che, dato un

rilevamento in una certa posizione, determina come tale misurazione influisce sullo stato di occupazione. Il secondo è l'applicazione del teorema di Bayes, per regolare il modo in cui l'occupazione delle celle viene aggiornata ad ogni nuovo campione temporale. Grazie al filtraggio Bayesiano è infatti possibile migliorare il rilevamento di ostacoli e ridurre i falsi allarmi, dando una descrizione probabilistica della cella più accurata aggiornandone ricorsivamente il valore.

Per ragioni computazionali e di calcolo numerico, è pratica comune utilizzare la cosiddetta rappresentazione log-odds di $P(\mathbf{m}|z_{1:t}, x_{1:t})$ durante la formulazione dell'algoritmo di costruzione della OGM, per ottenere risultati più stabili e sfruttare l'efficienza dell'additività ricorsiva nel dominio logaritmico. La funzione log-odds esprime la probabilità di un determinato evento A come il logaritmo del rapporto tra la sua probabilità di successo $P(A)$ e la sua probabilità di fallimento $(1 - P(A))$. La log-odds di $P(\mathbf{m}|z_{1:t}, x_{1:t})$ è quindi definita come segue:

$$l_t(m_i|z_{1:t}, x_{1:t}) = \log \left(\frac{P(m_i|z_{1:t}, x_{1:t})}{1 - P(m_i|z_{1:t}, x_{1:t})} \right) \quad (2.7)$$

Dalla definizione (2.7) e dall'ipotesi di indipendenza delle celle è possibile ricavare facilmente la probabilità di occupazione a posteriori della cella i -esima:

$$P(m_i|z_{1:t}, x_{1:t}) = 1 - \frac{1}{1 + e^{l_t(i)}} \quad (2.8)$$

Avendo presupposto un ambiente statico, ovvero che non cambi nel tempo, è possibile stimare l'occupazione di ciascuna cella utilizzando un filtro di Bayes binario. Ciò permette di ricavare la formula ricorsiva di aggiornamento della probabilità di occupazione della cella i -esima nel tempo, come dimostrato in [9]:

$$l_t(m_i|z_{1:t}, x_{1:t}) = l_t(m_i|z_t, x_t) + l_{t-1}(m_i|z_{1:t-1}, x_{1:t-1}) - l_0 \quad (2.9)$$

dove il secondo termine è legato alla probabilità di occupazione della cella fino all'istante $(t - 1)$, ovvero il termine ricorsivo. Il terzo termine,

la costante l_0 , indica lo stato di occupazione log-odds iniziale della cella:

$$l_0 = \log \left(\frac{P(m_i = 1)}{P(m_i = 0)} \right) \quad (2.10)$$

Tipicamente si assume che inizialmente $P(m_i = 1) = 0,5$ e $P(m_i = 0) = 0,5$ se non viene fornito alcun precedente sulla scena, e di conseguenza si ha $l_0 = 0$.

Occorre infine specificare che il primo termine $l_t(m_i|z_t, x_t)$ della (2.9) rappresenta l'ISM e determina in che modo viene aggiornata la cella all'istante corrente t , tenendo conto delle caratteristiche specifiche del sensore.

Utilizzando la formula iterativa (2.9) è quindi possibile implementare un algoritmo di aggiornamento del valore delle celle molto efficiente, che opera solo somme e sottrazioni sfruttando la rappresentazione log-odds. Si riporta di seguito un esempio di impostazione dell'algoritmo.

```

for (ogni misurazione nel tempo  $t = 1:T$ ) do
  for (ogni cella  $m_i$ ) do
    if ( $m_i$  interessata dalla misurazione  $z_t$ ) then
       $l_{t,i} = l_{t-1,i} + ISM(m_i, z_t, x_t) - l_0$ 
    else
       $l_{t,i} = l_{t-1,i}$ 
    end if
  end for
end for

```

È noto che le condizioni ambientali, la natura degli ostacoli osservati o persino le imperfezioni nella progettazione del sensore introducono rumore nelle misurazioni. Comunemente questa incertezza è tradotta in una distribuzione probabilistica chiamata *Sensor Model* (SM) [10], [11]. La SM fornisce la probabilità di ottenere un valore specifico dal rilevamento del sensore sapendo che la cella è libera oppure occupata da un target:

$$P(z_t | m_i = 1) \quad (2.11)$$

$$P(z_t | m_i = 0) \quad (2.12)$$

Quindi, crea un collegamento tra il mondo fisico e l'uscita del sensore. La funzione SM può essere costruita sperimentalmente oppure analiticamente tenendo conto della precisione del sensore indicata sulla sua scheda tecnica.

Per esempio si consideri un SM in cui i dati acquisiti dal radar consistono in un angolo θ e una distanza r per ogni target individuato nel campo visivo del sensore. Inoltre, si considerino angolo e distanza indipendenti tra loro. Con queste ipotesi il modello del sensore radar può essere descritto come:

$$P(\theta_t | m_i) = \frac{1}{\sqrt{2\pi}\sigma_\theta} e^{-\frac{1}{2}\left(\frac{\theta_t - \mu_\theta}{\sigma_\theta}\right)^2} \quad (2.13)$$

$$P(r_t | m_i) = \frac{1}{\sqrt{2\pi}\sigma_r} e^{-\frac{1}{2}\left(\frac{r_t - \mu_r}{\sigma_r}\right)^2} \quad (2.14)$$

dove θ ed r sono entrambi affetti da rumore gaussiano con valor medio μ_θ e μ_r , e varianza σ_θ^2 e σ_r^2 , rispettivamente.

Per dedurre le probabilità di occupazione a posteriori, è necessario invece produrre un ISM partendo da un certo SM. L'ISM esprime quindi la probabilità che una cella sia occupata data una rilevazione del sensore:

$$P(m_i = 1 | z_t) \quad (2.15)$$

e si lega al Sensor Model attraverso il teorema di Bayes, che applicato in questo caso risulta:

$$P(m_i = 1 | z_t) = \frac{P(z_t | m_i = 1)P(m_i = 1)}{P(z_t)} \quad (2.16)$$

dove $P(m_i = 1)$ rappresenta la probabilità di occupazione già associata alla cella i fino all'istante precedente la misura z_t . Alla prima iterazione dell'algoritmo ricorsivo si ha quindi $P(m_i = 1) = 0.5$ per ogni i .

Un'utile proprietà di questo approccio è che propaga naturalmente la precisione di rilevamento alla valutazione dell'occupazione, così l'incertezza del SM viene convertita in un'incertezza equivalente per l'ISM. Una delle funzioni ISM più comunemente utilizzata per i dati radar è la distribuzione Gaussiana [6]. Nel caso specifico del sensore radar è molto utile considerare incertezza e plausibilità delle misurazioni per aumentare l'accuratezza dell'aggiornamento della mappa. L'utilizzo di una Gaussiana segue infatti l'assunzione che la misurazione sia influenzata da rumore locale di varia natura.

Una volta aggiornata la OGM utilizzando tutte le misure temporali raccolte, si risale alla probabilità di occupazione a posteriori di ciascuna cella grazie alla (2.8). La mappa finale si ottiene sottoponendo le celle a una soglia: se la probabilità di occupazione è inferiore alla soglia, la cella sarà considerata non occupata, altrimenti sarà considerata occupata da un target.

2.2 Mappe a griglia di ampiezza

Un differente approccio alla mappatura dell'ambiente utilizzando dati radar è rappresentato dagli algoritmi di mapping a griglia d'ampiezza (*Amplitude Grid-Map* – AGM), trattati qualitativamente in [5] e [8]. Oltre alla localizzazione del bersaglio, le mappe a griglia di ampiezza ricavano informazioni utilizzando il RCS (*radar cross section*) del target. Poiché l'ampiezza del segnale riflesso ricevuto dal radar è legata al valore della RCS, essa è correlata alla capacità di riflessione dei bersagli e quindi al tipo di materiali di cui essi sono costituiti. Seguendo questo principio una mappa a griglia di ampiezza può ad esempio distinguere target metallici e non metallici.

Un metodo per costruire la griglia m dalle singole acquisizioni del radar potrebbe essere quello di scegliere come valore per ogni cella della mappa l'ampiezza del segnale ricevuto che ha rilevato la minor distanza dal radar in quella specifica cella. Questa scelta porta alla massima risoluzione di posizione effettiva, in quanto minore è la

distanza tra radar e target minore è l'incertezza di rilevazione. I valori della generica cella m_i sulla griglia dipenderanno solo da una singola misurazione del radar, e avranno quindi la stessa incertezza della singola misurazione.

Un secondo tipo di approccio prevede invece che il valore della cella m_i della mappa a griglia di ampiezza in posizione (x, y) sia la media pesata di tutte le ampiezze $A_{x,y}(k)$ rilevate dalle osservazioni radar di questa cella fino all'istante temporale t , con $0 \leq k \leq t$. Le singole osservazioni sono pesate in base all'inverso della distanza $r_{x,y}(k)$ del target ad ogni t . Grazie a questa pesatura le misurazioni a minore distanza dal radar avranno più influenza sulla costruzione della griglia finale rispetto alle misurazioni a distanza maggiore. Considerato che la griglia è costruita su una media di più misurazioni, per la legge della propagazione dell'incertezza si ha che l'incertezza sull'ampiezza totale è minore di quella su una misurazione singola [8].

A causa dei diversi approcci di modellazione, i due metodi di costruzione della griglia presentati possiedono qualità differenti. Le caratteristiche di contorno e di posizione dei target nelle OGM sono generalmente più chiare, mentre le AGM possono caratterizzare maggiormente le diverse tipologie di ostacoli.

Un esempio di questi due metodi di mappatura in ambito stradale è mostrato nella figura 2.1. Partendo da sinistra sono poste a confronto una immagine fotografica, una AGM ed una OGM che raffigurano lo stesso ambiente, assegnando ad ogni colore utilizzato un livello di occupazione, da blu (libero) a rosso (occupato). Si osserva che la AGM rileva sia i bordi stradali come spazio occupato (ovvero la vegetazione, quindi ostacoli "deboli") che lo spazio libero oltre questi, mentre la OGM individua come occupati i lati della strada senza però riuscire ad ottenere informazioni oltre ad essi; infatti lo spazio circostante viene questa volta modellato come sconosciuto.

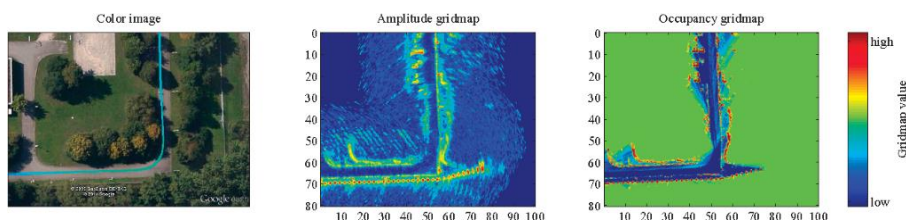


Figura 2.1 Confronto tra OGM ed AGM [8]

2.3 Mapping collaborativo

Negli ultimi anni, sempre più studi hanno rivolto particolare attenzione alla ricerca di metodi per migliorare i risultati del rilevamento e della classificazione di oggetti basati sui dati radar. I concetti di “*Data fusion*” e “*Collaborative mapping*” nascono a questo scopo, e si riferiscono a tecniche di combinazione di dati provenienti da molteplici sensori [5]. L’obiettivo principale di questi sistemi è quello di ottenere valutazioni più specifiche rispetto a quelle ottenibili usando un singolo sensore indipendente, riducendo l’incertezza dei dati rilevati e migliorandone la robustezza e l’accuratezza.

In generale, parlando di *data fusion*, ci si riferisce all’unione di dati provenienti da sensori di diverso tipo. Una comune soluzione di rilevamento basato sulla fusione di differenti sensori nel campo automotive è la combinazione di radar e informazioni visive. Questa configurazione sfrutta contemporaneamente le ricche informazioni semantiche delle immagini e le informazioni su posizione e movimento provenienti dal radar, per ottenere mappe ambientali più dettagliate e costruire una buona base per applicazioni che prevedono un processo decisionale, come ad esempio la guida autonoma.

Nel caso in esame si parlerà di fusione di dati provenienti da più sensori dello stesso tipo, il radar FMCW. Un singolo radar è in grado di generare fino a diverse centinaia di rilevamenti per frame. Tuttavia è noto che la nuvola di punti radar non sia regolare e significativamente più rada rispetto ad esempio a quella prodotta da un sensore LIDAR. Inoltre, la diffusione delle onde elettromagnetiche alla frequenza operativa standard del radar automobilistico è molto sensibile agli angoli di incidenza, al materiale dell’ostacolo e alla rugosità della superficie. Ciò fa sì che la nuvola di punti abbia una qualità non elevata e appaia differente da un fotogramma all’altro. Pertanto, molto spesso si incrociano dati provenienti da radar multipli posizionati secondo una configurazione nota, per aumentare la qualità della mappa ambientale ottenuta. Un approccio standard consiste nel costruire una mappa a griglia di occupazione (OGM) dall’accumulo di più frame di dati acquisiti da diverse angolazioni, come descritto in [11]. Con lo scorrere del tempo lo spazio viene sondato dai diversi sensori, le informazioni si

sovrappongono progressivamente, e le probabilità di occupazione delle celle vengono aggiornate dinamicamente. Le griglie di occupazione sono molto popolari nella robotica probabilistica e sono spesso utilizzate come base per il tracciamento, la percezione, la mappatura, la localizzazione e pianificazione del percorso. Le OGM sono infatti tipicamente impiegate per la fusione di dati da diversi sensori, poiché la l'aggiornamento delle mappe in termini di probabilità di occupazione si presta bene al mapping collaborativo.

Nei radar automobilistici, un tipico campo visivo azimutale (*field of view* - FOV) per ottenere dati di buona qualità è di circa 80°-100°. Pertanto, per ricavare una mappatura completa dell'ambiente, sul veicolo dovrebbero essere montati più radar con diversi orientamenti azimutali. Inoltre, ampie sovrapposizioni tra i FOV dei radar vicini introducono ridondanze nei rilevamenti e aiutano a coprire le distanze ravvicinate in modo più denso, migliorando così la robustezza complessiva del sistema di rilevamento e percezione. L'approccio OGM ha un potenziale particolarmente grande nel contesto di una rete di sensori radar per i seguenti motivi:

- Il filtraggio Bayesiano migliora il rilevamento di target e riduce i falsi allarmi nei radar vicini.
- L'accumulo di dati multi-frame consente di filtrare le incongruenze e produrre una mappa di probabilità di occupazione stabile.

Un radar 3D in genere produce più rilevamenti puntiformi per bersaglio. Ogni rilevamento è caratterizzato da una posizione 3D descritta in coordinate polari sferiche (r, θ, φ) , poiché il radar assegna a ciascuna misurazione i valori di range, azimuth ed elevazione. Tenendo conto delle incertezze relative ai tre diversi dati, i più avanzati modelli di costruzione della OGM riducono le probabilità di occupazione lungo direzioni che non producono rilevamenti, mentre le aumentano nelle celle in cui viene individuato un target [11]. Questo doppio meccanismo si dimostra particolarmente utile in un contesto multi-sensore e multi-frame, poiché i falsi rilevamenti da parte di radar diversi e/o in momenti diversi non sono correlati e quindi possono essere efficacemente mitigati.

3 Radar Infineon BGT60ATR24C

I moderni sensori per il riconoscimento dell'ambiente sono spesso basati su sistemi radar, nel caso specifico su radar FMCW. Tali sistemi comprendono diversi blocchi e richiedono un alto livello di integrazione in un singolo chipset.

Il dispositivo BGT60ATR24C della Infineon è un sensore radar per automobili che opera nel range di frequenze tra 58 e 62 GHz, consente il funzionamento FMCW con banda ultra-larga (4GHz) e opera a corto raggio con una massimo range operativo intorno ai 2,5 m. La configurazione del frame e le impostazioni tecniche del sensore sono gestite tramite la *Sensing GUI*, un'interfaccia digitale per facilitare l'interazione con l'utente.

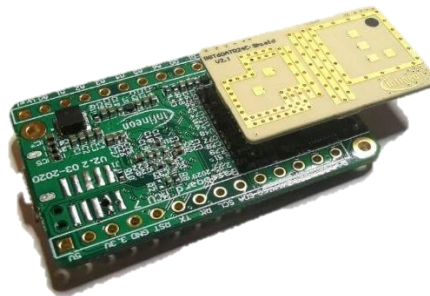


Figura 3.1 Infineon radar BGT60ATR24C

La funzionalità principale di BGT60ATR24C consiste nel trasmettere il segnale a onda continua modulata in frequenza (FMCW) tramite uno dei due antenne di trasmissione (TX) e ricevere i segnali di eco riflessi dal target sulle quattro antenne di ricezione (RX). Ogni percorso del ricevitore include un filtro in banda base, un VGA (*Video Graphic Array*) e un convertitore da analogico a digitale (ADC). L'output digitalizzato viene memorizzato e trasferito a un host esterno, un'unità microcontrollore (MCU) o un processore applicativo (AP), per eseguire l'elaborazione del segnale radar.

Il chipset è stato progettato per implementare principalmente le seguenti potenziali applicazioni:

- Front-end radar per il rilevamento dei gesti
- Radar FMCW ad alta risoluzione
- Operazioni di rilevamento a corto raggio

3.1 Parametri di controllo

L'interfaccia digitale permette all'utente di configurare il frame utilizzato e controllare i parametri legati al chirp e alle modalità di trasmissione e ricezione del segnale. In particolare il produttore fornisce anche la *Matlab Radar System API*, che consente di leggere o modificare i parametri del sensore ed acquisire i dati direttamente in ambiente Matlab.

In tabella 1 sono elencati i principali parametri configurabili tramite l'interfaccia Matlab.

PARAMETRI GENERALI	
<i>num_samples_per_chirp</i>	Campioni acquisiti per ogni shape (chirp) trasmessa da una antenna TX e ricevuta da una antenna RX. Valori accettati [1...4095].
<i>num_chirps_per_frame</i>	Numero di chirp trasmessi da ogni TX per ogni frame.
<i>adc_samplerate_hz</i>	Frequenza di campionamento del convertitore analogico/digitale.
<i>lower_frequency_kHz</i>	Frequenza minima dei chirp trasmessi.
<i>upper_frequency_kHz</i>	Frequenza massima dei chirp trasmessi.

<i>bgt_tx_power</i>	Potenza trasmessa (per entrambe le TX) su una scala [0-31].
<i>rx_antenna_mask</i>	Maschera binaria che stabilisce quali delle 4 antenne RX sono attive. $15_{(10)} = 1111_{(2)} = 4$ RX attive.
<i>tx_mode</i>	Valore pari a 0 se è attiva TX1, pari a 1 se è attiva TX2.
<i>if_gain_Db</i>	Amplificazione della IF per entrambe le antenne TX. Valori accettati [18...60] [dB].
<i>frame_end_delay_s</i>	Ritardo di fine frame, prima dell'acquisizione del frame successivo. Valori accettati [0...27380] [s].
<i>shape_end_delay_s</i>	Ritardo di fine shape, prima dell'acquisizione della shape successiva. Valori accettati [0...27380] [s].
<i>radar_height</i>	Altezza del radar in metri.
<i>threshold</i>	Soglia di rilevamento [dB].
$BW = (upper_frequency_kHz - lower_frequency_kHz) * 1e3$	Larghezza di banda dei chirp trasmessi.
$r_res = c0 / (2 * BW)$	Range resolution, con $c0 = 3 \cdot 10^8$ [m/s] velocità luce.
$range_max = r_res * num_samples_per_chirp / 2$	Distanza radiale massima misurabile [m].
$Fc = (upper_frequency_kHz - lower_frequency_kHz) * 1e3 / 2$	Frequenza di centro banda dei chirp.
$lambda = c0 / Fc$	Lunghezza d'onda dei chirp [m] .

Tabella 1 Parametri di controllo

L'acquisizione diretta tramite Matlab permette di elaborare comodamente i dati grezzi per ottenere le principali informazioni riguardo i target intercettati. Per ogni ostacolo individuato è possibile ricavare:

- *Range* – distanza radiale tra radar e oggetto, in metri.
- *Azimuth* – angolo compreso tra l'asse x e la retta che unisce il radar alla proiezione del punto individuato (target) sul piano orizzontale.
- *Elevation* – angolo di elevazione in gradi rispetto al piano orizzontale del radar.

Le tre variabili vengono poi rappresentate tramite grafici cartesiani (figura 4.2), dove ogni punto rappresenta un rilevamento.

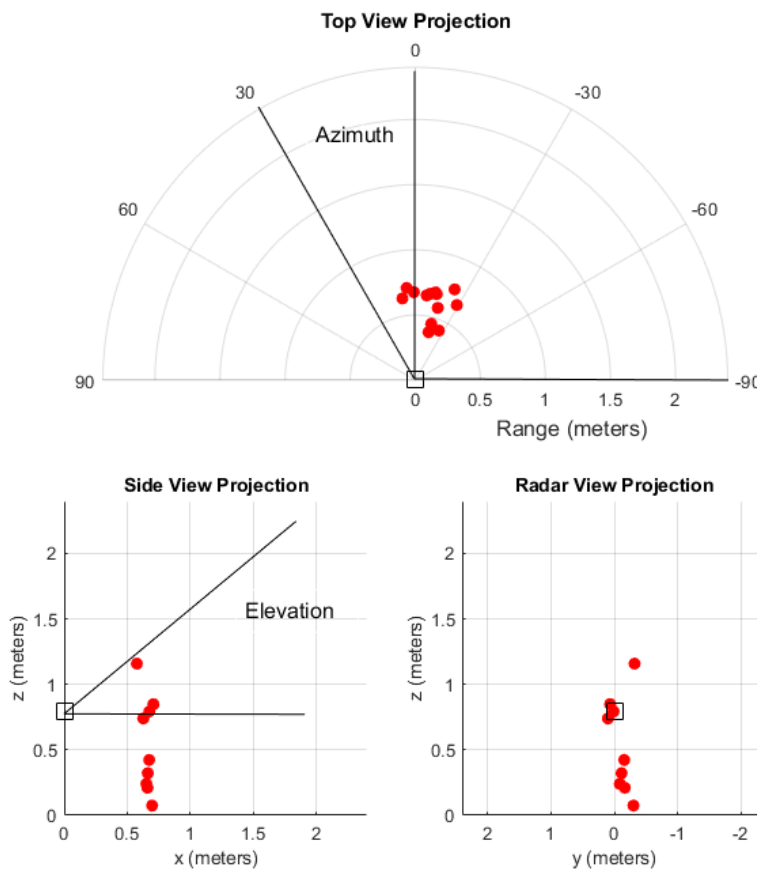


Figura 3.2 Rappresentazione dei target rilevati: *Range, Azimuth ed Elevation*

3.2 Configurazione

Nel caso in esame si utilizza una configurazione che prevede l'utilizzo di tutte le antenne disponibili. Lo shape set (figura 3.4) è composto da due up-chirp, il primo trasmesso da TX1 e il secondo da TX2. In particolare ognuna delle due antenne TX trasmette alternativamente 10 up-chirp per frame, ottenendo quindi un frame composto da 20 shape (ovvero 10 shape set). Tutte e quattro le antenne RX sono attive in ricezione ed ognuna di esse acquisisce 128 campioni per ogni shape del frame. In figura 3.3 è mostrato il grafico dei segnali ricevuti [mV] nel tempo [μs]. Ogni riquadro riporta i quattro segnali ricevuti dalle quattro RX (con colore diverso per ogni RX) per ogni shape trasmessa da TX1 e TX2; due riquadri consecutivi rappresentano uno dei 10 shape set.

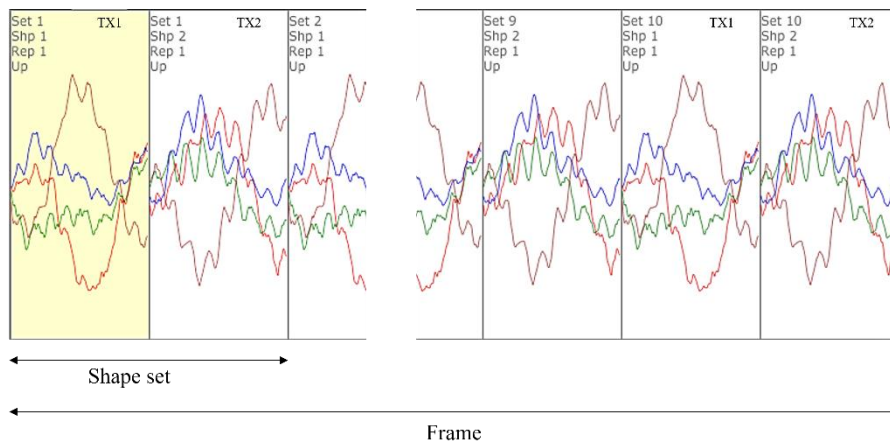


Figura 3.3 Configurazione frame (immagine da Sensing GUI)

I chirp trasmessi hanno una frequenza minima di 58 MHz e una massima di 62 MHz, ottenendo così una banda di 4 GHz. Di conseguenza la range resolution vale:

$$r_{res} = \frac{c_0}{2 \cdot BW} \cong 0,0375 \text{ m} \quad (3.1)$$

e il range massimo misurabile:

$$r_{max} = \frac{r_{res} \cdot 128}{2} \cong 2,4 \text{ m} \quad (3.2)$$

con 128 campioni per chirp.

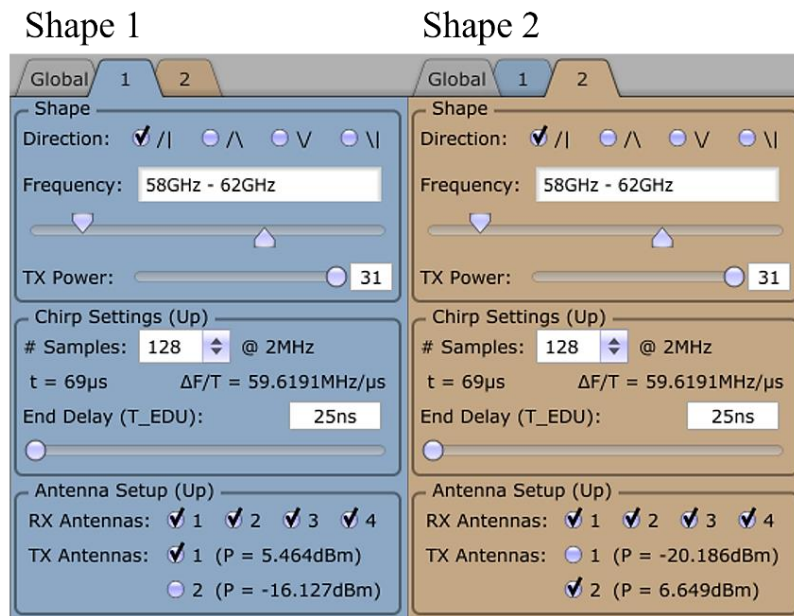


Figura 3.4 Configurazione shape set (immagine da Sensing GUI)

Si impostano inoltre i seguenti valori:

<i>if_gain_Db</i>	40	[dB]
<i>bgt_tx_power</i>	31	-
<i>adc_samplerate_hz</i>	2000000	[Hz]
<i>frame_end_delay_s</i>	1	[s]
<i>shape_end_delay_s</i>	$150 \cdot 10^{-6}$	[s]

Tabella 2 Configurazione parametri

I dati “grezzi” raccolti vengono salvati in matrici tridimensionali “*RawData.mat*”, dove la terza dimensione rappresenta il numero di frame, le quattro colonne le antenne RX e le righe i campioni totali acquisiti durante il frame da ognuna delle quattro RX, come mostrato in figura 3.5 (sinistra).

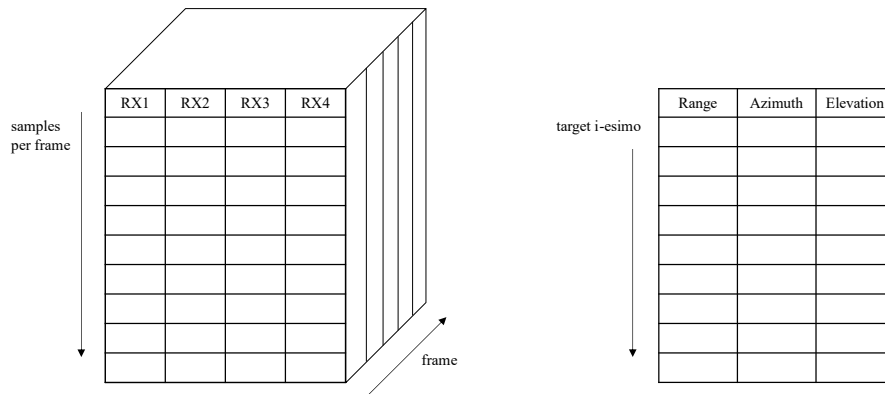


Figura 3.5 Matrice “*RawData*” a sinistra e matrice “*TargetData*” a destra

Una volta elaborati i dati utilizzando il codice Matlab fornito dal produttore si ottiene una seconda matrice “*TargetData.mat*” (figura 3.5, destra), questa volta bidimensionale, che per ogni target individuato (righe) riporta le variabili range, azimuth ed elevation corrispondenti (colonne).

La campagna sperimentale è stata svolta utilizzando la configurazione appena descritta, acquisendo dati in diversi contesti ambientali e quindi variando a seconda dei casi parametri come soglia, altezza e inclinazione del radar.

4 Campagna sperimentale

L'obiettivo del lavoro di tesi sperimentale è quello di applicare sul campo la teoria legata ai principi di funzionamento di un sistema radar e ed in seguito implementare un algoritmo di mapping a griglia di occupazione per uno scenario statico.

In particolare sono stati posizionati due sensori sulle maniglie anteriore e posteriore di un'auto (figura 4.1), in modo da avere misurazioni riguardanti lo stesso ambiente da due angolazioni diverse. Lo scopo finale è infatti quello di indentificare eventuali ostacoli presenti attorno all'automobile, creando una mappa 3D che rappresenti il target grazie alla fusione dei dati raccolti dai due radar.

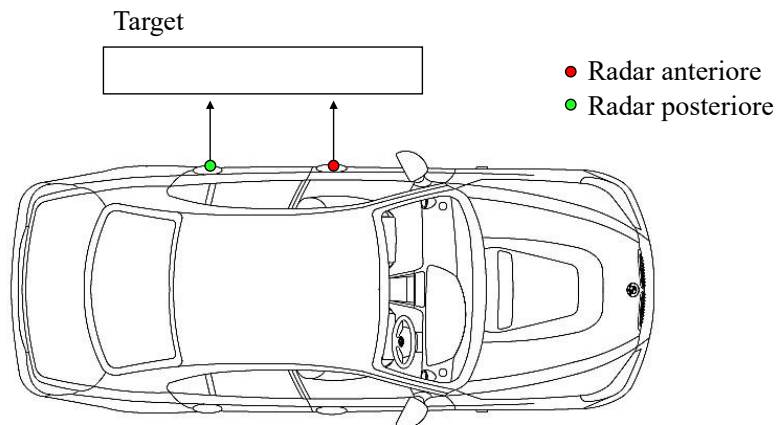


Figura 4.1 Rappresentazione grafica degli esperimenti in scenario reale

4.1 Esperimenti preliminari

Utilizzando il radar Infineon BGT60ATR24C a disposizione, sono stati eseguite inizialmente rilevazioni in un ambiente statico controllato volte a verificare le funzionalità del sensore. In figura 4.2 e 4.3 è

riportato il setup di misura utilizzato per svolgere gli esperimenti preliminari.



Figura 4.2 Scenario controllato per esperimenti preliminari

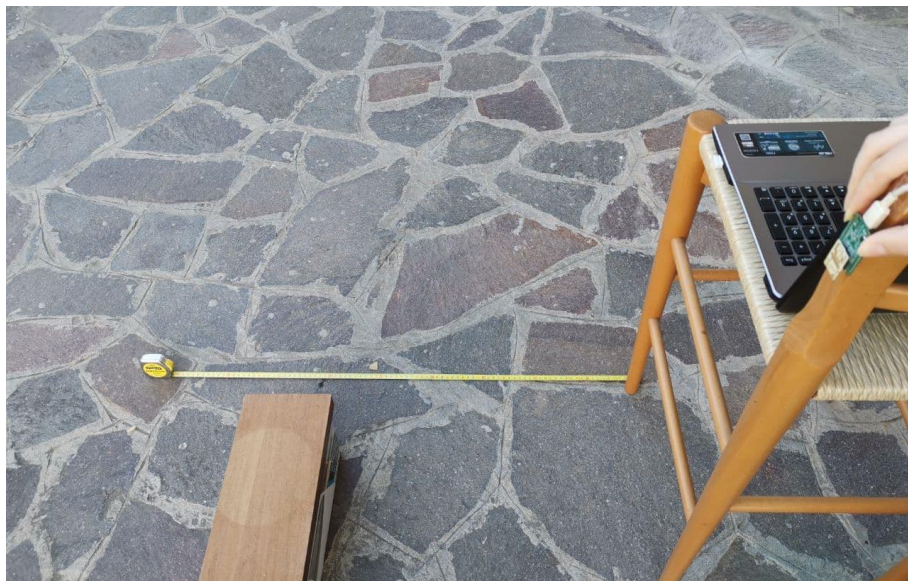


Figura 4.3 Scenario controllato per esperimenti preliminari

Si illustrano in seguito diversi scenari indagati, considerando un sistema di riferimento ortogonale (x, y, z) , come in figura 4.4. I parametri di controllo vengono impostati come descritto nel paragrafo 3.2, e la soglia di rilevamento è fissata a -55 dB.

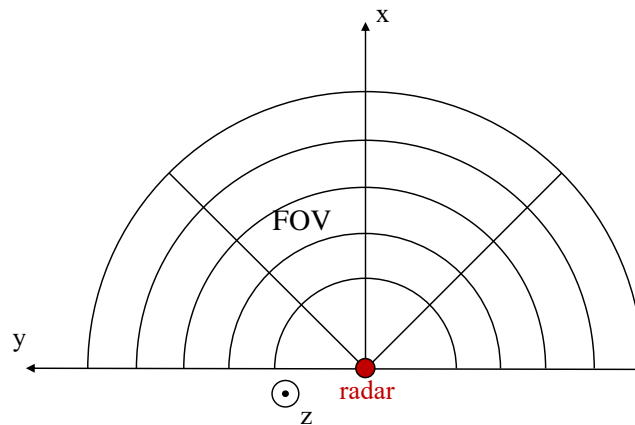


Figura 4.4 Sistema di riferimento

Ogni acquisizione è stata poi elaborata utilizzando il codice Matlab fornito, in modo da ottenere la rappresentazione grafica di range, azimuth ed elevation per ogni target rilevato. In seguito si riportano alcuni degli scenari preliminari indagati e i risultati ottenuti.

Scenario 1

Posizionamento del radar:

- altezza (asse z): 0.82 m
- coordinate (x, y) : $(0, 0)$ m
- inclinazione rispetto all'asse z : 0°

Posizionamento del target:

- altezza (asse z): 0.20 m
 - coordinate (x, y) : $(0.40, 0)$ m
-

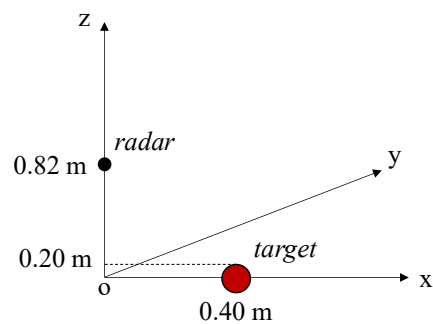


Figura 4.5 Scenario n° 1

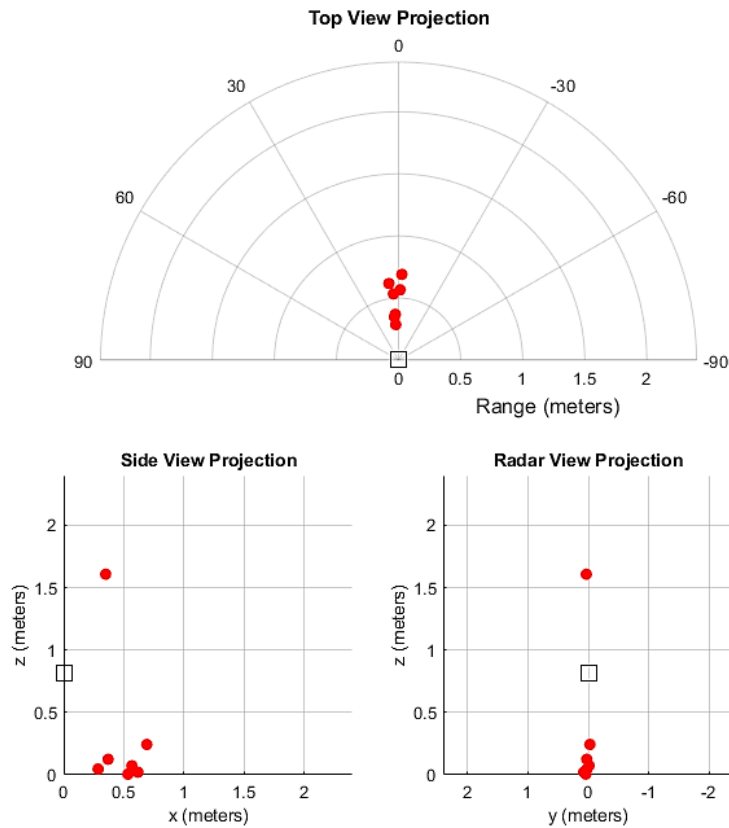


Figura 4.6 Scenario n° 1, risultati

In figura 4.5 è schematizzata la configurazione di radar e target indagata. I risultati della misurazione, che utilizza i parametri riportati sopra, sono illustrati in figura 4.6, tramite i tre grafici restituiti dal codice Matlab di acquisizione. La *Top View Projection* rappresenta la visione dall'alto, con il radar posto nell'origine del piano (x, y) , e permette di visualizzare sia il range radiale tra target e sensore che l'angolo azimutale in gradi. La *Side View Projection* raffigura invece il piano (x, z) , con il radar posto in coordinate $(x, radar_height)$; in questo caso è ben visibile l'angolo di elevazione in gradi, dove lo 0° di riferimento è ad altezza sensore. In fine si ha la *Radar View Projection*, ovvero il piano (y, z) , che propone il punto di vista del radar stesso.

In generale si osserva una buona corrispondenza tra l'ambiente reale e la misurazione del sensore, pur notando delle false rilevazioni sistematiche sopra il radar, a circa 1.5 m di altezza, riscontrate anche negli scenari seguenti. È probabile che queste siano dovute a riflessioni

d'ambiente indesiderate, ed essendo sistematiche non verranno considerate durante la creazione della mappa.

Scenario 2

Posizionamento del radar:

- altezza (asse z): 0.82 m
- coordinate (x, y) : (0, 0) m
- inclinazione rispetto all'asse z : 0°

Posizionamento del target:

- altezza (asse z): 0.10 m
- coordinate (x, y) : (1.5, 0) m

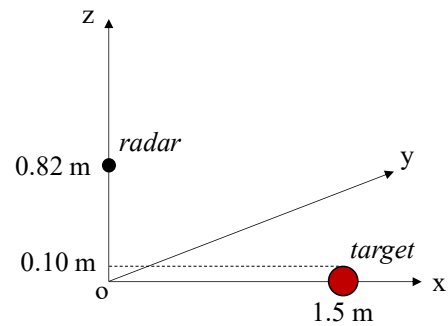


Figura 4.7 Scenario n° 2

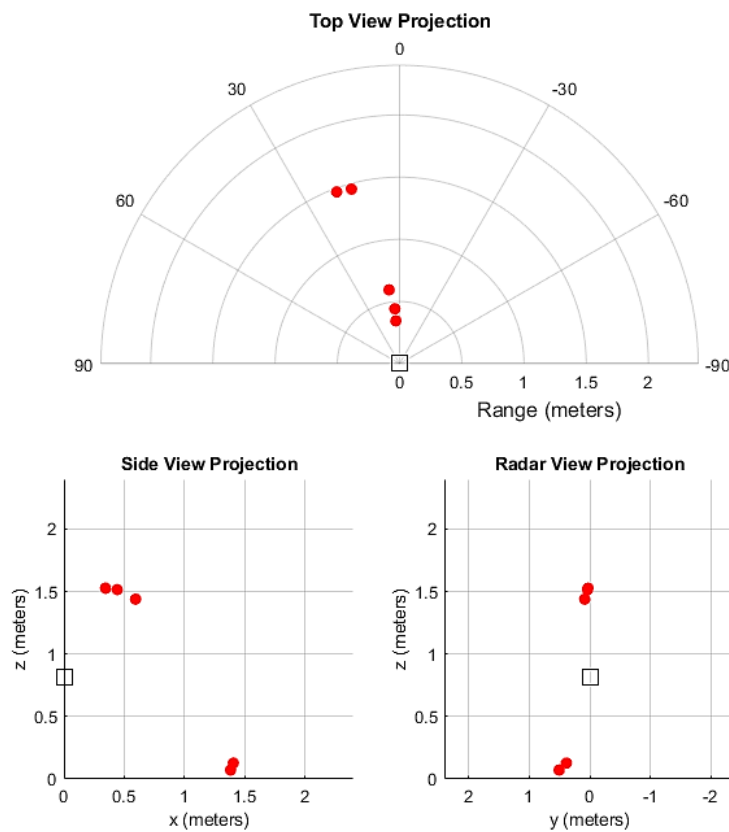


Figura 4.8 Scenario n° 2, risultati

Scenario 3

Posizionamento del radar:

- altezza (asse z): 0.82 m
- coordinate (x, y) : (0, 0) m
- inclinazione rispetto all'asse z : 0°

Posizionamento del target:

- altezza (asse z): 0.82 m
- coordinate (x, y) : (1.0, 0) m

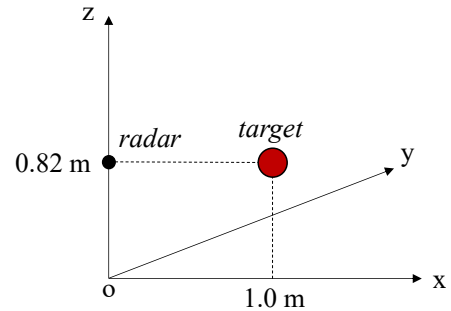


Figura 4.9 Scenario n° 3

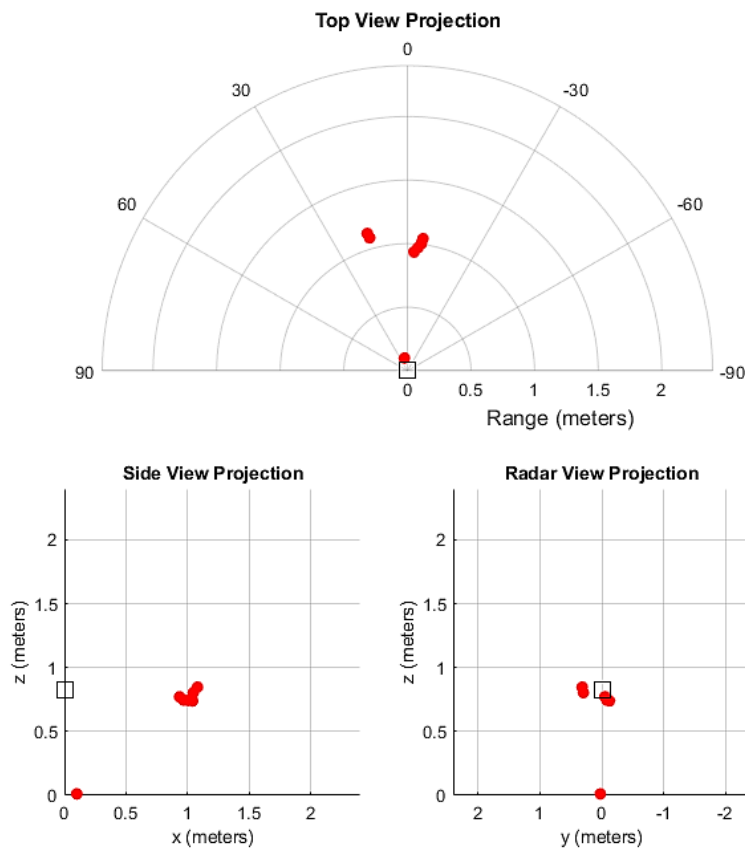


Figura 4.10 Scenario n° 3, risultati

4.2 Esperimenti in scenario reale

Una volta conclusi gli esperimenti preliminari utili per prendere confidenza con il sistema e a settare appropriatamente i parametri, si passa ad uno scenario statico reale. In particolare il sensore viene ora posizionato sulla maniglia della portiera di un'automobile, acquisendo dati in uno scenario urbano (figura 4.11).

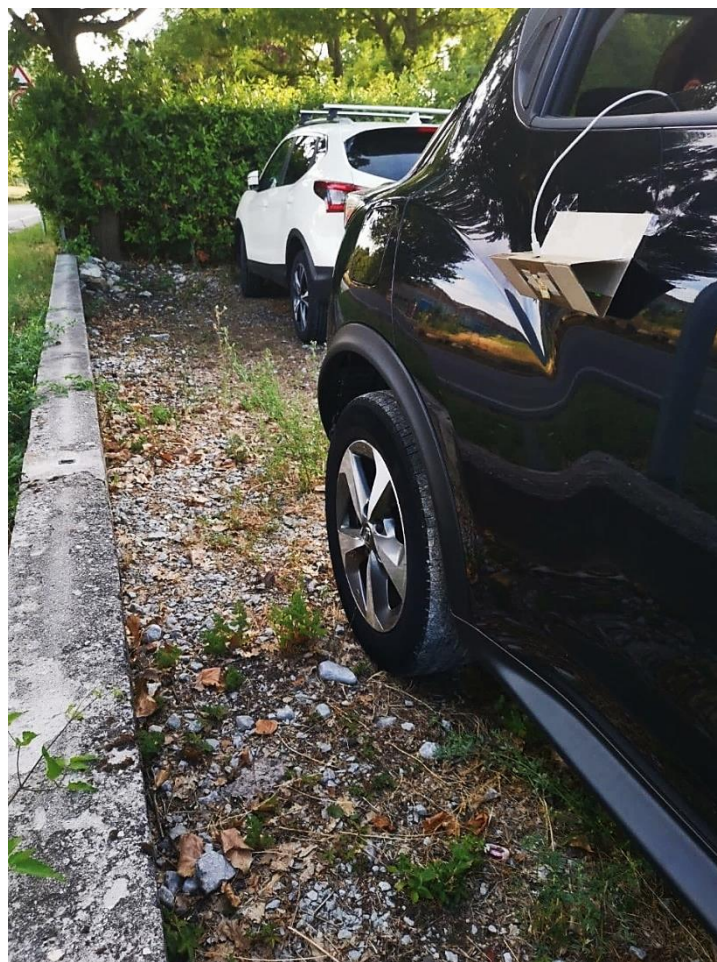


Figura 4.11 Esperimenti in uno scenario statico reale

Si esamina in questo modo una situazione reale in cui il radar svolge una potenziale applicazione di rilevamento di ostacoli statici nelle

vicinanze di un'automobile ferma. A questo scopo il sensore, posizionato in corrispondenza della maniglia, viene inclinato di circa 25° verso il basso, in modo da ottenere le prestazioni massime nel riconoscimento di target a terra che potrebbero ostacolare l'apertura della portiera. Questa inclinazione viene poi compensata da una post-elaborazione in Matlab della matrice “*TargetData*”, ottenendo così grafici in coordinate assolute.

La configurazione dei parametri di controllo del radar utilizzata per le rilevazioni sperimentali è quella in tabella 3.

<i>if_gain_Db</i>	40	[dB]
<i>bgt_tx_power</i>	31	-
<i>adc_samplerate_hz</i>	2000000	[Hz]
<i>frame_end_delay_s</i>	1	[s]
<i>shape_end_delay_s</i>	$150 \cdot 10^{-6}$	[s]
<i>rx_antenna_mask</i>	4	-
<i>threshold</i>	-50	[dB]
<i>num_samples_per_chirp</i>	128	-
<i>num_chirps_per_frame</i>	10	-
<i>lower_frequency_kHz</i>	58	[MHz]
<i>upper_frequency_kHz</i>	62	[MHz]

Tabella 3 Configurazione parametri

Nella parte seguente del capitolo si illustrano alcuni dei diversi scenari indagati e i risultati ottenuti elaborando i dati in Matlab.

Scenario reale 1

Posizionamento del radar:

- altezza (asse z): 0.80 m
- coordinate (x, y) : (0, 0) m
- inclinazione rispetto all'asse z : 25°

Target rilevato:

- altezza (asse z): 0.50 m
- coordinate (x, y) : (0.50, 0) m

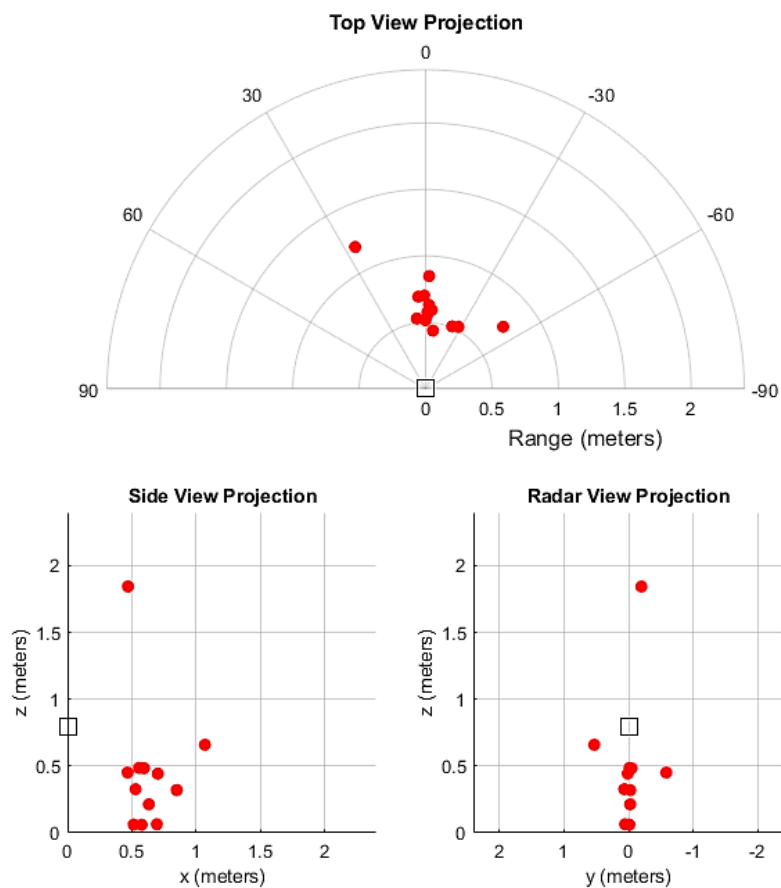


Figura 4.12 Scenario reale n° 1

In questo primo scenario reale si nota abbastanza chiaramente la rilevazione del profilo di un muretto alto circa 50 cm e distante circa 50 cm dall'auto. Si riscontra inoltre il disturbo sistematico nell'area sopra il radar.

Scenario reale 2

Posizionamento del radar:

- altezza (asse z): 0.80 m
- coordinate (x, y) : (0, 0) m
- inclinazione rispetto all'asse z : 25°

Target rilevato:

- altezza (asse z): 0.40 m
- coordinate (x, y) : (0.30, 0) m

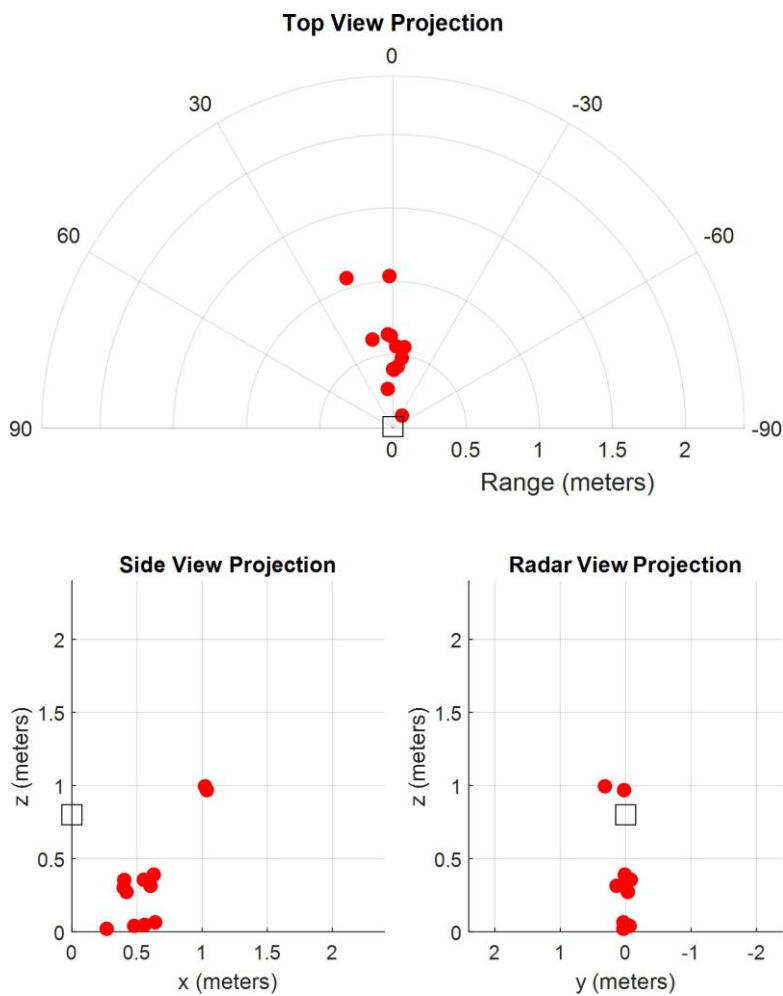


Figura 4.13 Scenario reale n° 2

Anche in questo caso si osserva abbastanza chiaramente il profilo di un muretto questa volta alto circa 40 cm e distante circa 30 cm dall'auto.

Scenario reale 3

Posizionamento del radar:

- altezza (asse z): 0.80 m
- coordinate (x, y) : (0, 0) m
- inclinazione rispetto all'asse z : 25°

Target rilevato:

- altezza (asse z): 0.20 m
- coordinate (x, y) : (0.50, 0) m

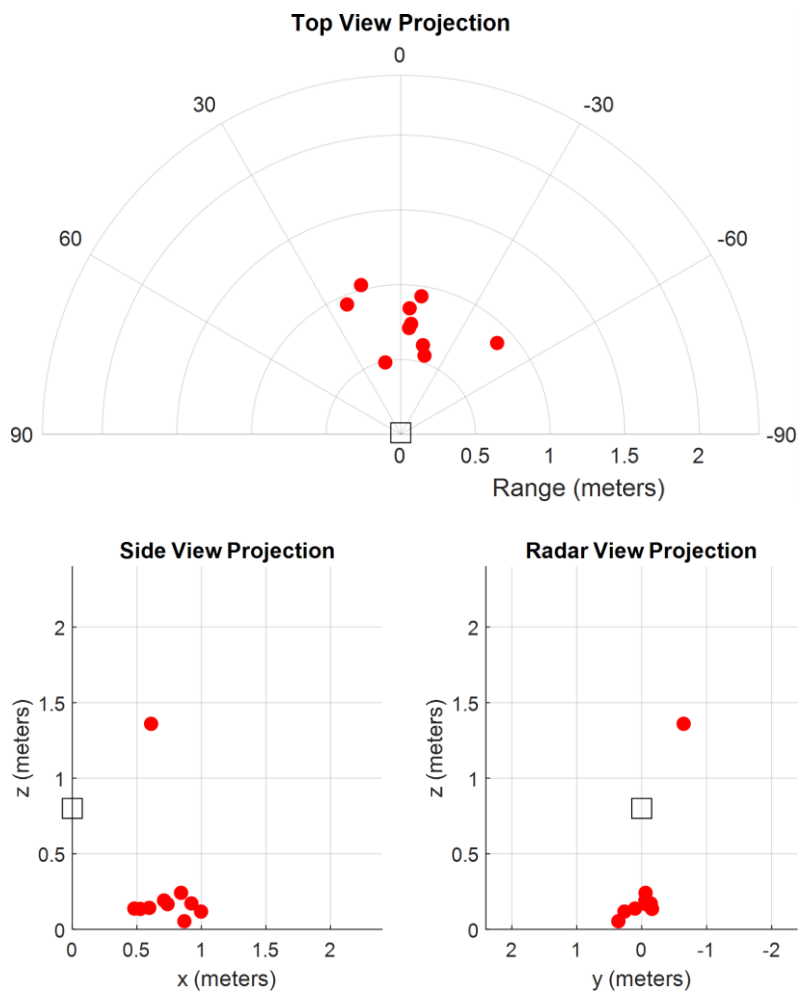


Figura 4.14 Scenario reale n° 3

Anche in quest'ultimo scenario si distingue il profilo di un marciapiede alto 20 cm e distante 50 cm dall'auto. La rilevazione resta comunque disturbata dalla falsa rilevazione sistematica nella parte alta.

Infine si svolgono alcune rilevazioni posizionando il sensore prima sulla maniglia della portiera anteriore poi su quella posteriore, senza cambiare scenario. In questo modo si acquisiscono dati riguardanti il medesimo target da due punti di vista differenti, per poi combinarli in post-elaborazione (nel capitolo 5) ed ottenere una mappa con precisione maggiore.

Scenario reale 4

Radar anteriore:

- coordinate (x, y, z) : $(0, 0, 0.80) m$
- inclinazione rispetto asse z : 25°

Radar posteriore:

- coordinate (x, y, z) : $(0, 0.75, 0.80) m$
- inclinazione rispetto asse z : 25°

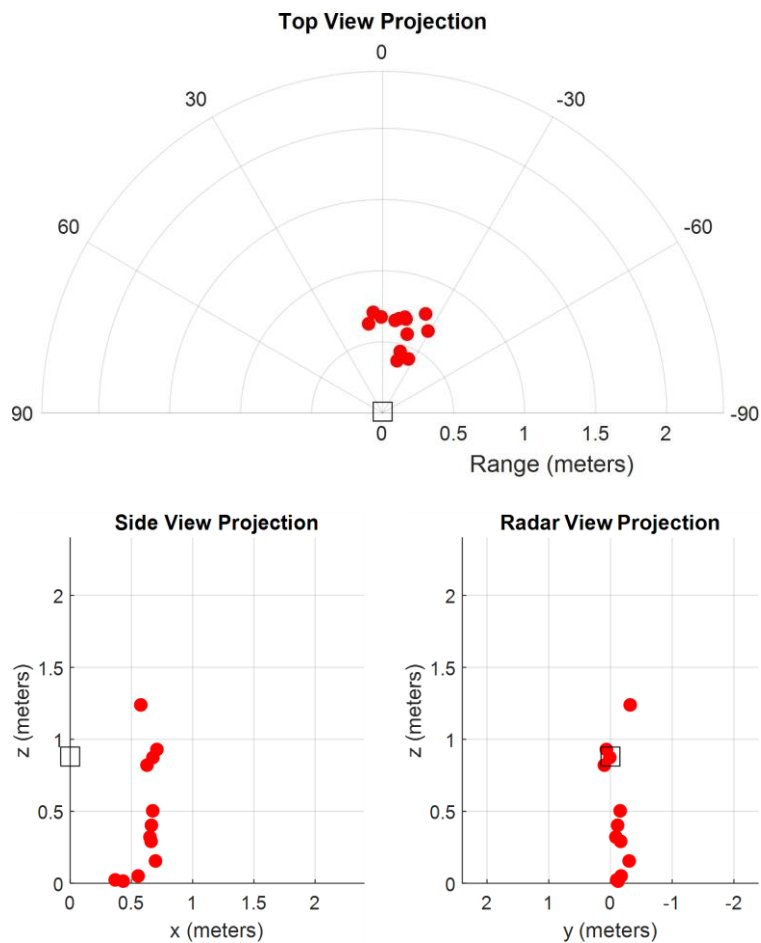


Figura 4.15 Scenario reale n° 4, da portiera anteriore

Scenario reale 5

Radar anteriore:

- coordinate (x, y, z) : $(0, 0, 0.80) m$
- inclinazione rispetto asse z : 25°

Radar posteriore:

- coordinate (x, y, z) : $(0, 0.75, 0.80) m$
- inclinazione rispetto asse z : 25°

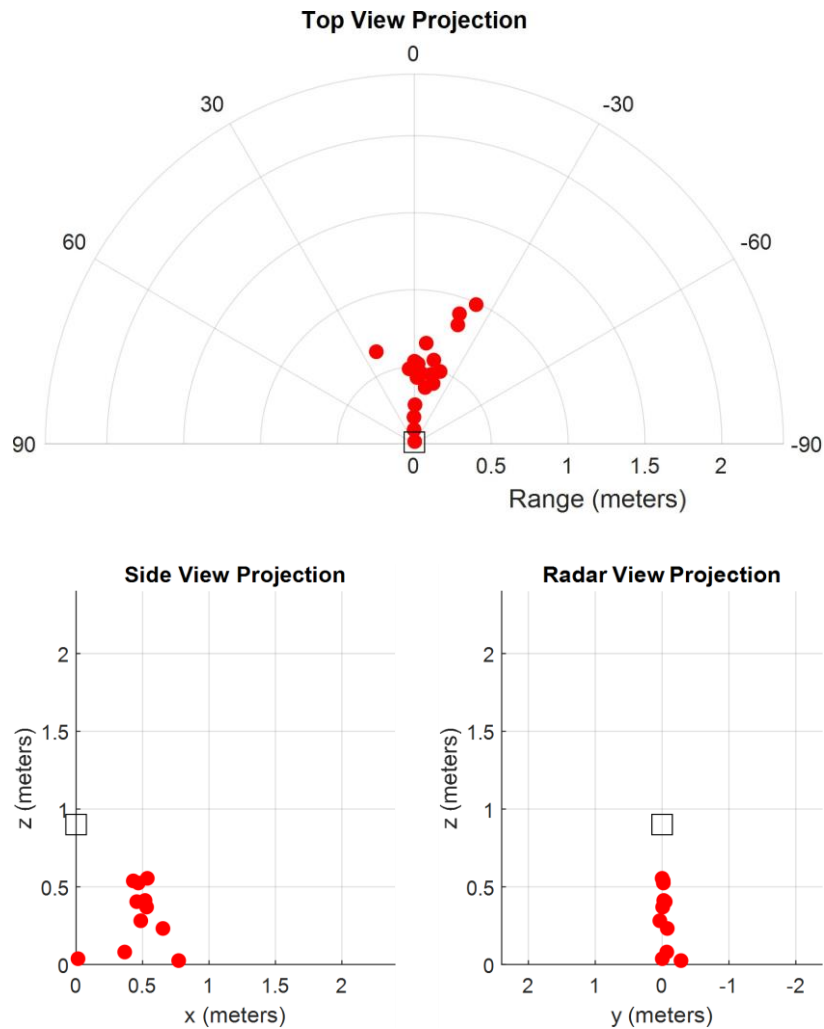


Figura 4.16 Scenario reale n° 5, da portiera anteriore

In questi ultimi due casi si hanno quindi due rilevazioni riguardo il medesimo target, grazie ai due radar posizionati alla stessa altezza e distanti tra loro 75 cm sull'asse y .

Nello scenario reale 4 entrambi i sensori rilevano chiaramente il muro accanto all'automobile, distante circa 60 *cm*. Si riportano in figura 4.15 i grafici ottenuti dalla rilevazione del radar posto sulla portiera anteriore.

Infine, nello scenario reale 5, si osserva distintamente il profilo di un muretto alto circa 55 *cm* e distante 45 *cm* dall'auto; sono illustrati in figura 4.16 i grafici ottenuti dalla rilevazione del radar sulla portiera anteriore.

5 Elaborazione dati e risultati

In seguito all'acquisizione dei dati sperimentali l'obiettivo è quello di elaborarli e produrre una mappa 3D probabilistica a griglia di occupazione, che rappresenti l'ambiente osservato dai due radar posizionati sul perimetro dell'automobile.

A questo scopo si è sviluppato un codice di mapping 3D in ambiente Matlab utilizzando il *Navigation toolbox*, che permette agevolmente di suddividere uno spazio tridimensionale in celle volumetriche, caratterizzate ciascuna da una probabilità di occupazione. La funzione `occupancyMap3D(res,'OccupiedThreshold',occ)` inizializza una mappa tridimensionale consentendo di impostare la risoluzione (*res*, in celle per metro) e la soglia di occupazione *occ*; le celle che alla fine della costruzione della mappa risultano avere una probabilità superiore alla soglia vengono plottate come occupate.

La suddetta mappa viene creata utilizzando una struttura dati *octree* che consente aggiornamenti efficienti e probabilistici dello spazio occupato e libero mantenendo al minimo il consumo di memoria. Un octree è una struttura dati gerarchica per la suddivisione spaziale in 3D [12]; ogni nodo di un octree rappresenta lo spazio contenuto in un volume cubico, tipicamente chiamato *voxel*.

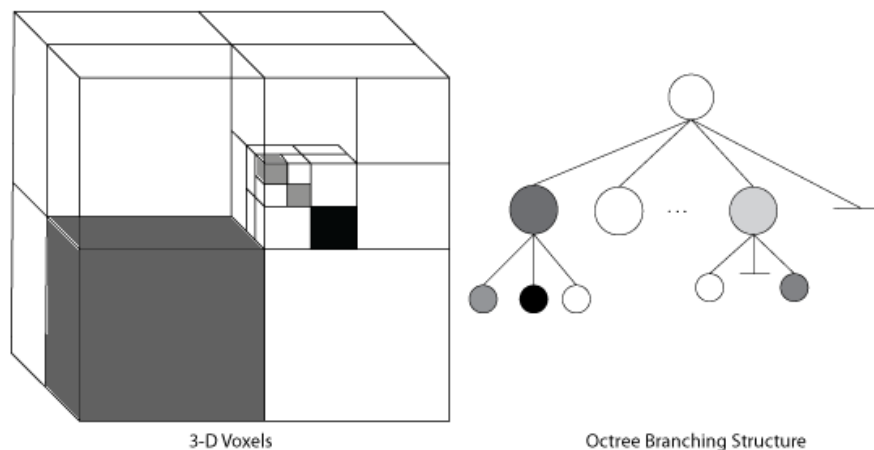


Figura 5.1 Octree data structure [12]

Questo volume viene suddiviso ricorsivamente in otto sotto-volumi fino a raggiungere una data dimensione minima di voxel, che determina la risoluzione dell'octree (figura 5.1). Questa rappresentazione ad albero offre la massima flessibilità per quanto riguarda l'area mappata e la risoluzione. L'octree gestisce quindi arbitrariamente l'ambiente ed espande le sue dimensioni dinamicamente in base agli input di osservazione. Se un determinato volume è rilevato come occupato, viene inizializzato il nodo corrispondente nell'octree. Per quanto riguarda l'aggiornamento delle probabilità di ogni cella, per ogni misurazione vengono creati esplicitamente volumi liberi nell'area tra il sensore e il punto finale misurato, mentre le aree non inizializzate rappresentano lo spazio sconosciuto. L'ISM di default utilizzato dal Navigation Toolbox per aggiornare il valore di ogni volume attraversato dal raggio visivo tra radar e target è il seguente:

$$l_t(m_i|z_t) = l_{occ} \quad \text{se il raggio viene riflesso dal volume} \quad (5.1)$$

$$l_t(m_i|z_t) = l_{free} \quad \text{se il raggio attraversa il volume} \quad (5.2)$$

dove l_{occ} e l_{free} sono valori log-odds utilizzati come valori di ISM nella (2.9). Nel caso in esame si ha $l_{occ} = 0.85$ e $l_{free} = -0.4$, corrispondenti rispettivamente alle probabilità 0.7 e 0.4 (tramite la relazione (2.8)). L'uso dello stato di occupazione come etichetta della cella consente rappresentazioni compatte dell'octree: se tutti i figli di un nodo hanno lo stesso stato di occupazione possono essere “potati”. Questo porta ad una sostanziale riduzione del numero di nodi che devono essere mantenuti attivi nell'albero, e quindi ad un utilizzo molto efficiente dello spazio di memoria.

L'aggiornamento probabilistico delle celle tramite ISM è implementato tramite la funzione `updateOccupancy(map3D,xyz,obs)` del toolbox, che utilizza i valori di occupazione *obs* per aggiornare i punti di coordinate *xyz* nella mappa *map3D*. I valori *obs* di default sono quelli riportati in (5.1) e (5.2); in questo studio sono stati però utilizzati due diversi approcci, descritti nel seguito del capitolo, che prevedono uno un valore di *obs* pari a 0.65 per ogni punto individuato dal radar, l'altro un valore di *obs* diverso per ogni punto e basato sull'intensità della rilevazione dello stesso.

Nei sistemi radar, si deve spesso far fronte al rumore d'ambiente e ad acquisizioni che cambiano temporaneamente a causa di riflessioni indesiderate. Per questo lo stato di occupazione deve essere modellato probabilisticamente ad ogni nuova acquisizione temporale, in questo caso applicando un algoritmo di mappatura a griglia di occupazione come quello illustrato nel capitolo 2.1 tramite *updateOccupancy*. La funzione combina quindi la compattezza degli octree con l'aggiornabilità e la flessibilità della modellazione probabilistica.

Nel caso in esame tutta la mappa viene inizializzata a probabilità 0.5, per supporre di non avere informazioni a priori sull'ambiente. In seguito, vengono integrate probabilisticamente le misurazioni dei due radar distinti in un unico ambiente. Impostando come soglia di occupazione in valore di probabilità di 0.65 solo le celle che al termine dell'algoritmo saranno associate ad un valore maggiore o uguale alla soglia verranno rappresentate come occupate sul grafico.

Per completezza si producono inoltre due grafici 2D dei piani (x,y) e (x,z) , in cui ogni cella risulta colorata in base al corrispondente valore di occupazione su una scala $[0.001; 0.999]$, dove i valori maggiori o uguali a 0.65 rappresentano le celle occupate, ovvero i target rilevati.

Le acquisizioni sperimentali sono state svolte utilizzando il codice Matlab di acquisizione, che traduce i dati grezzi in una matrice *TargetData* contenente le coordinate di ogni target individuato. In seguito questi dati sono stati rielaborati per costruire la OGM, tramite il codice che implementa il mapping 3D (Appendice B).

Per produrre la mappa finale sono stati testati due diversi approcci di aggiornamento della probabilità delle celle, illustrati in seguito.

Metodo 1

I sensori radar sono tipicamente affetti da un alto numero di false rilevazioni dovute a riflessioni o rumore d'ambiente.

Il primo approccio utilizzato per la costruzione della mappa 3D prevede che la "scrematura" dei punti rumorosi avvenga durante la prima elaborazione, tramite il codice di acquisizione. Regolando una

soglia direttamente durante le rilevazioni sperimentali si ottiene quindi una matrice *TargetData* che contiene le coordinate dei punti rilevati con una potenza superiore ad essa, punti che quindi si suppongono occupati da un target con una probabilità superiore a 0.5. Quando questi dati vengono poi utilizzati nel codice di mapping, supponendo che ogni punto rilevato sia potenzialmente occupato si utilizza come valore unico di aggiornamento la probabilità di 0.85 nella funzione *updateOccupancy*.

Si riportano in seguito i risultati grafici ottenuti.

Scenario 1

Muro distante 60 cm, alto 150 cm.

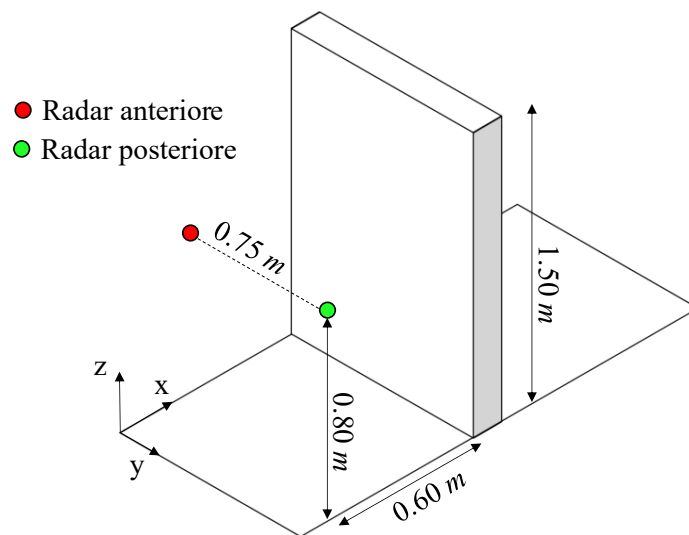


Figura 5.2 Rappresentazione grafica dello scenario n° 1

Nel seguito si riportano tutti i grafici ottenuti dall'elaborazione dei dati raccolti nel primo scenario (figura 5.2). Dalle figure (5.3) e (5.4) si distingue abbastanza chiaramente il profilo del muro individuato; nelle figure (5.5) e (5.6) si possono invece osservare le viste dall'alto e laterale, che confermano la corretta rilevazione del target. Le medesime considerazioni si possono applicare agli scenari 2 e 3 successivi.

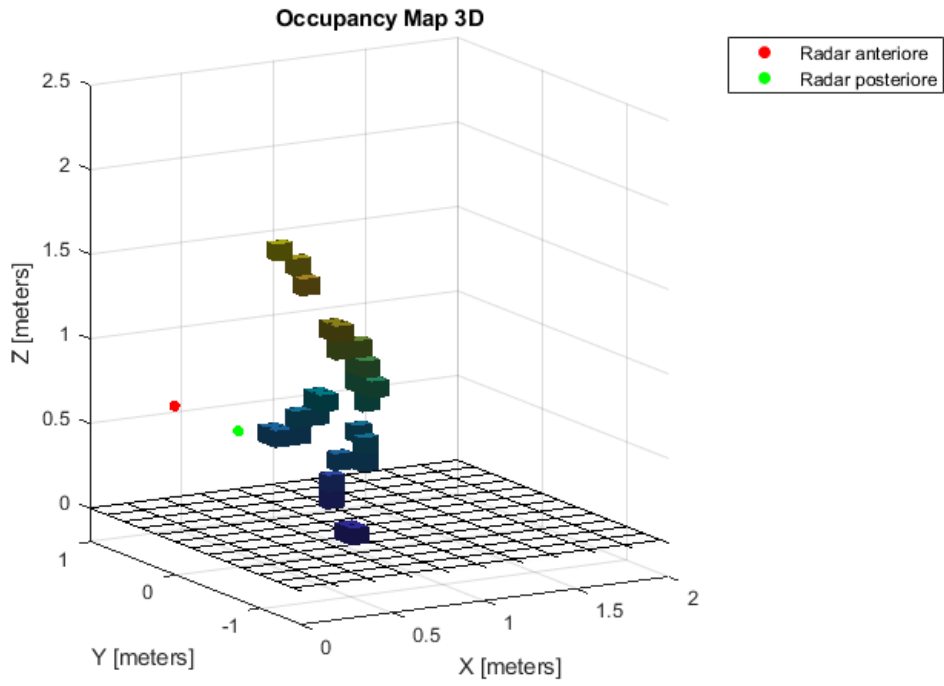


Figura 5.3 OGM 3D

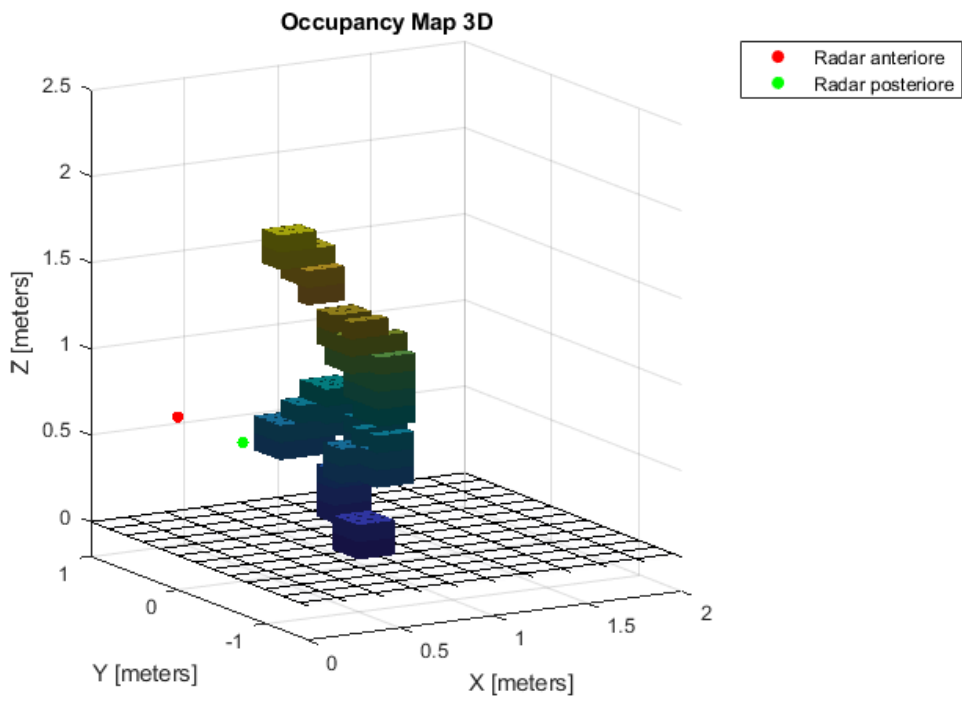


Figura 5.4 OGM 3D, inflate

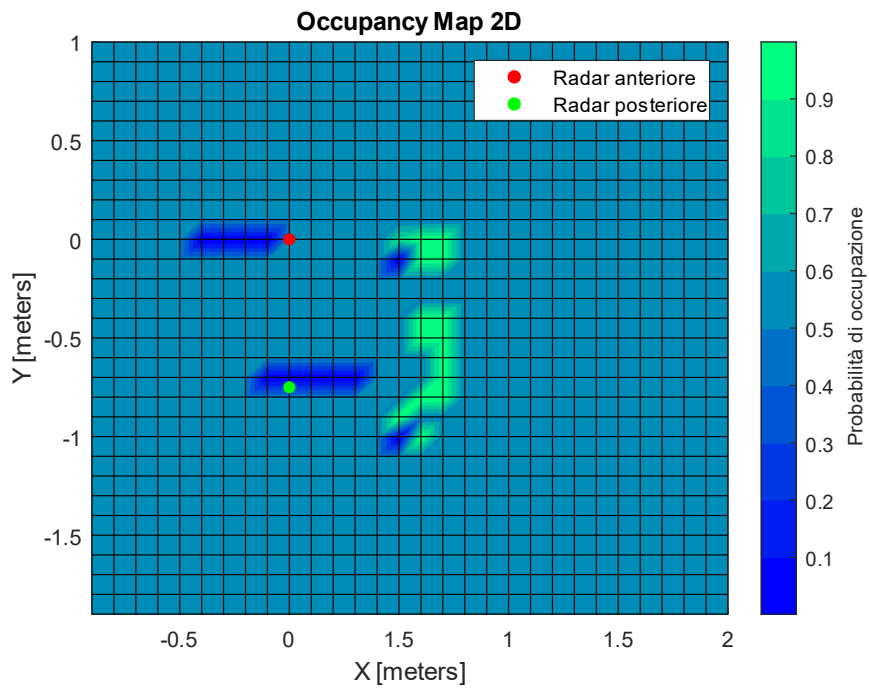


Figura 5.5 OGM 2D, piano (x, y)

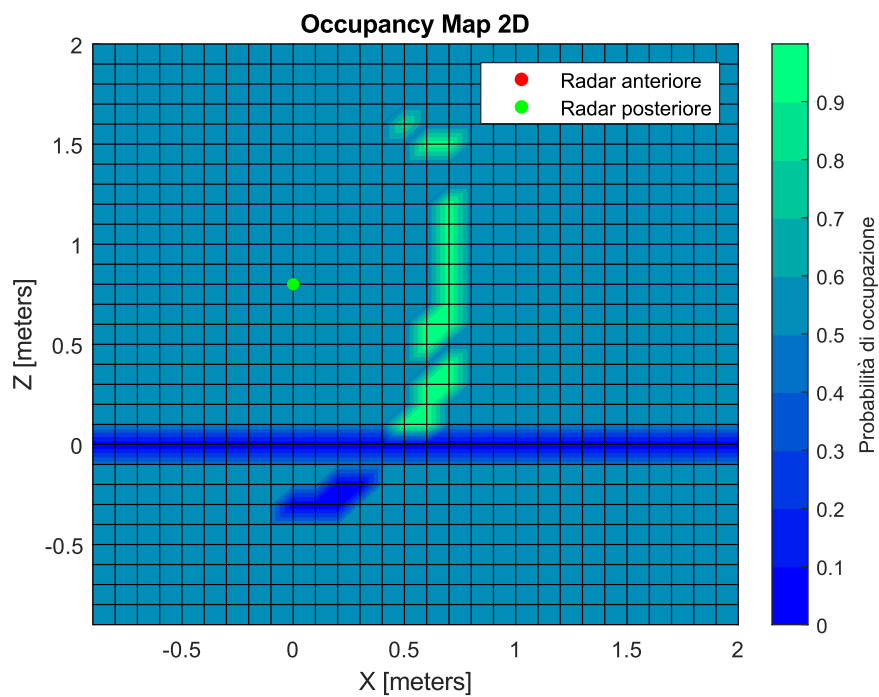


Figura 5.6 OGM 2D, piano (x, z). Secondo questo punto di vista il radar anteriore si trova dietro il radar posteriore.

Scenario 2

Muretto distante 50 cm, alto 50 cm.

- Radar anteriore
- Radar posteriore

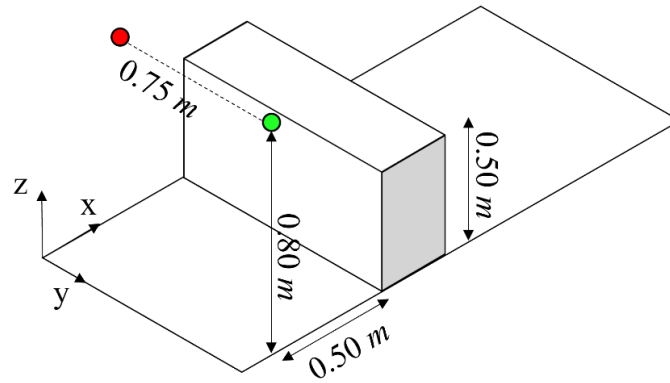


Figura 5.7 Rappresentazione grafica dello scenario n° 2

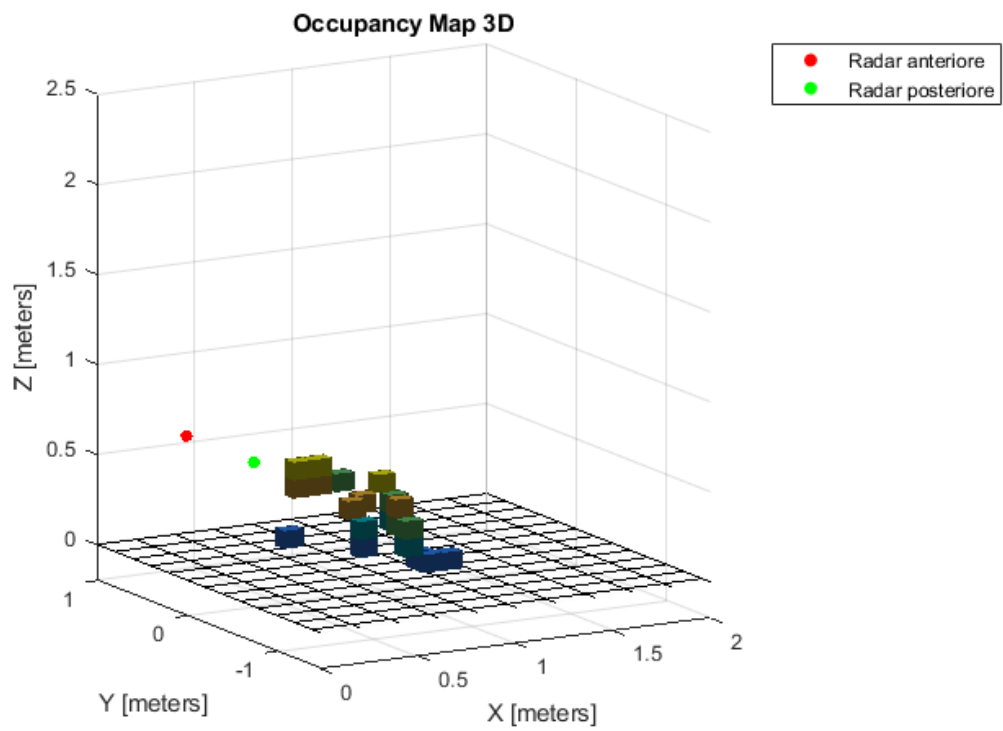


Figura 5.8 OGM 3D

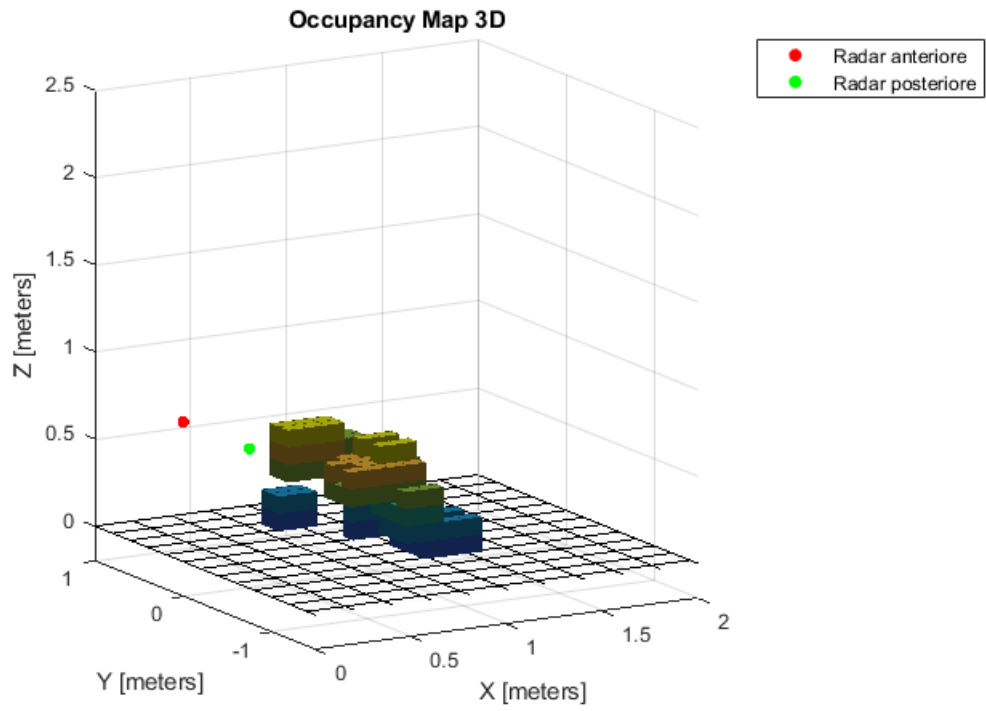


Figura 5.9 OGM 3D, inflate

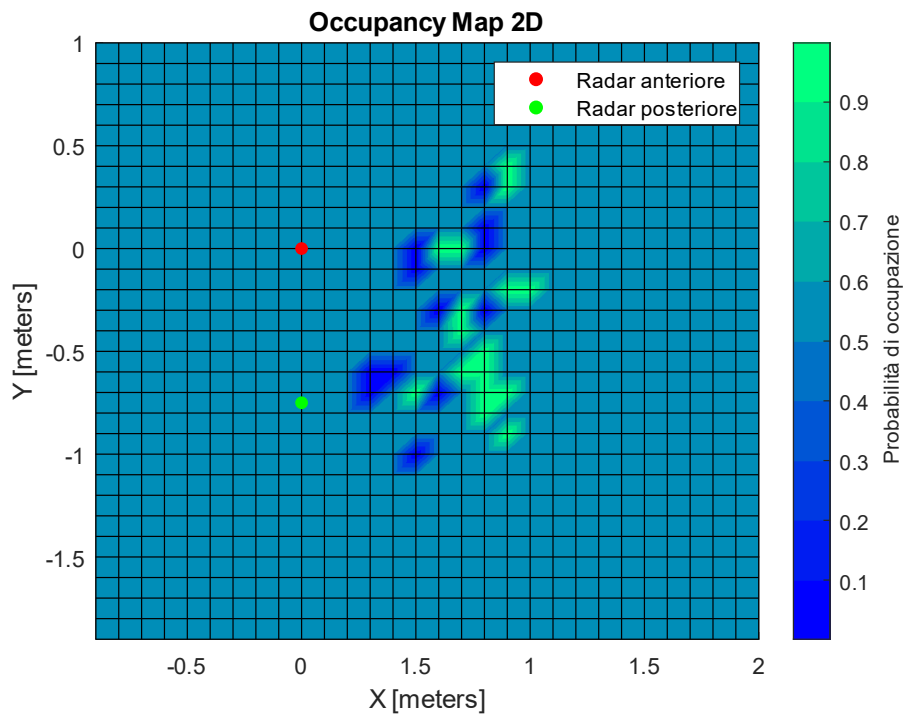


Figura 5.10 OGM 2D, piano (x, y)

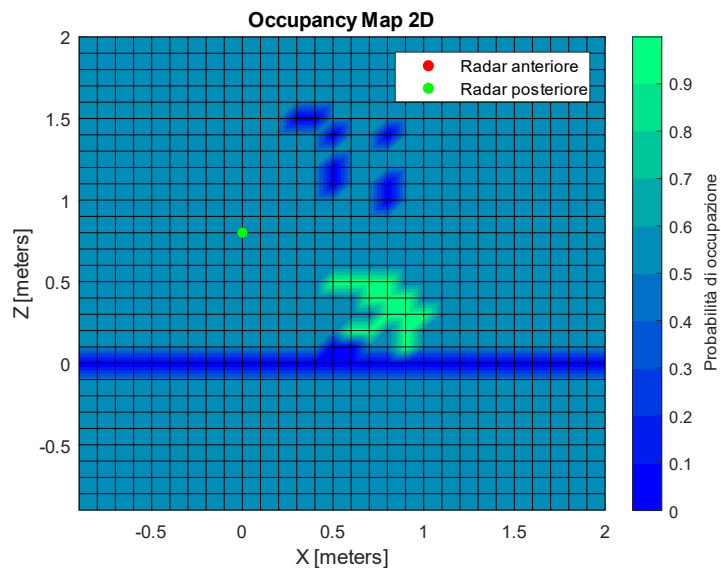


Figura 5.11 OGM 2D, piano (x, z). Secondo questo punto di vista il radar anteriore si trova dietro il radar posteriore.

Scenario 3

Marciapiede distante 45 cm, alto 30 cm.

- Radar anteriore
- Radar posteriore

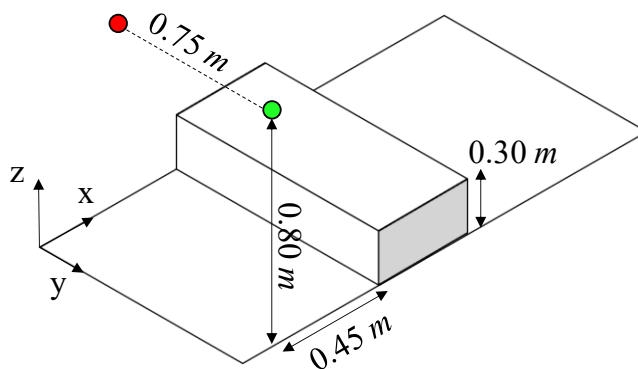


Figura 5.12 Rappresentazione grafica dello scenario n° 3

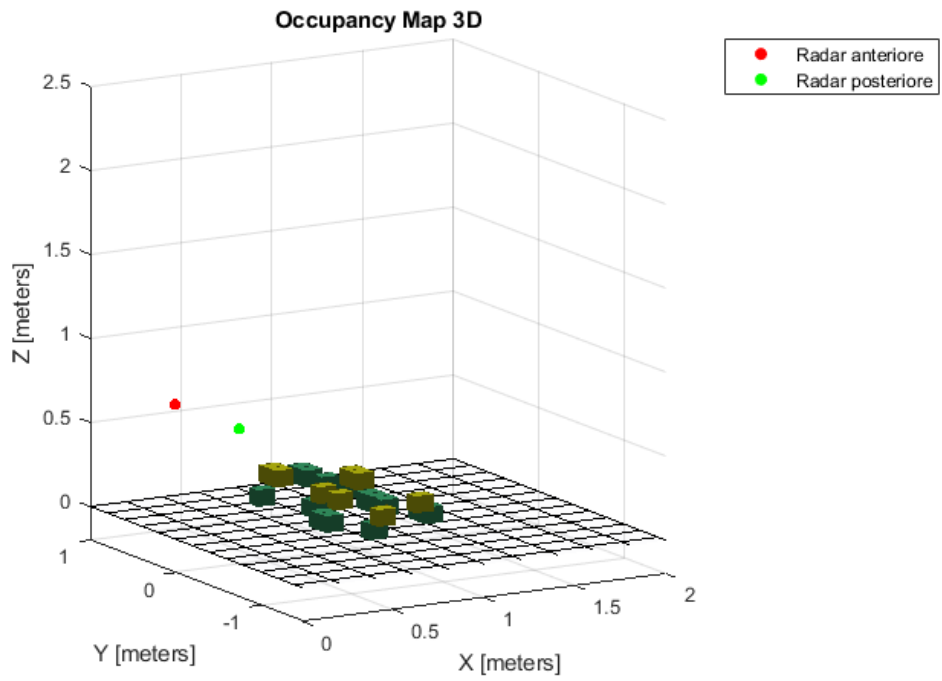


Figura 5.13 OGM 3D

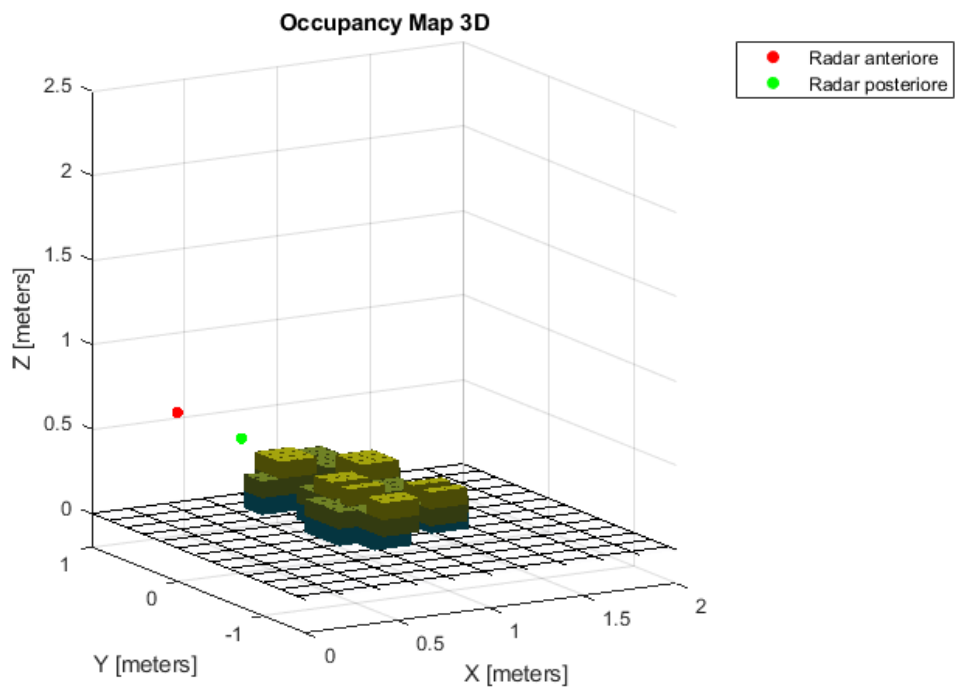


Figura 5.14 OGM 3D, inflate

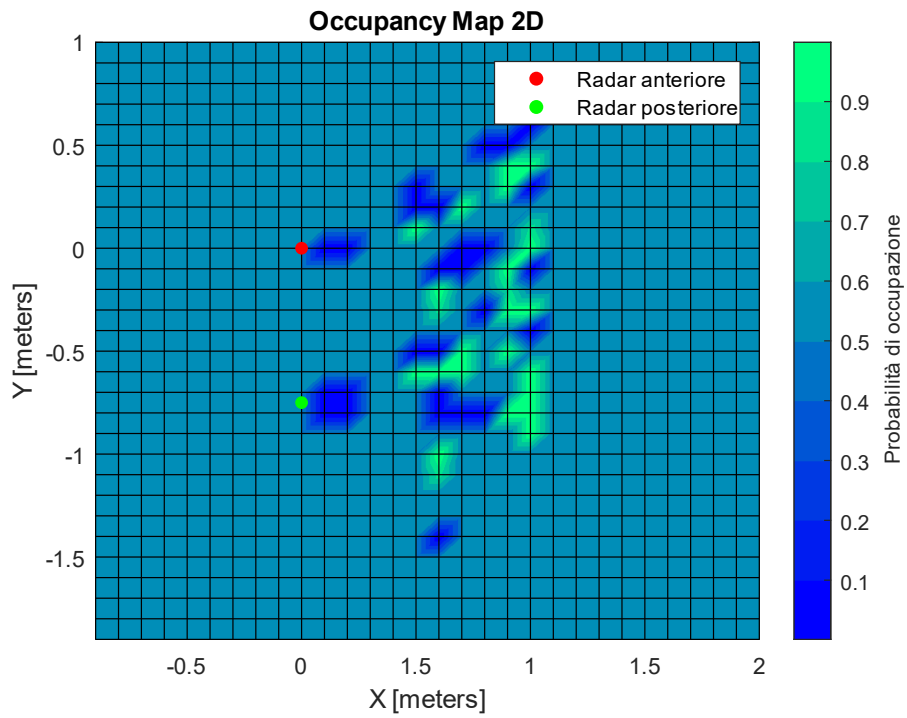


Figura 5.15 OGM 2D, piano (x, y)

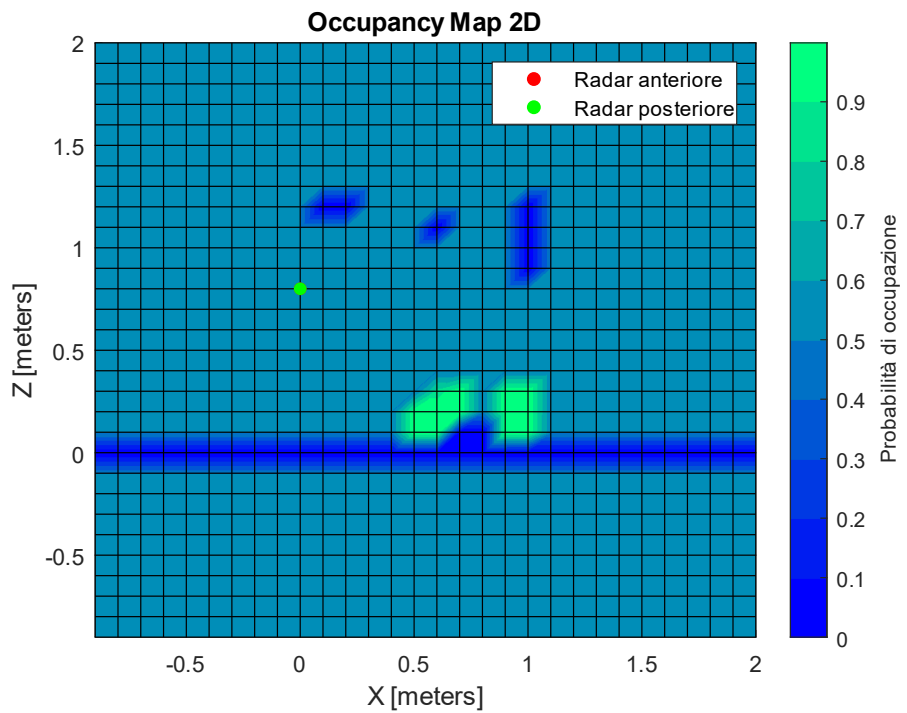


Figura 5.16 OGM 2D, piano (x, z). Secondo questo punto di vista il radar anteriore si trova dietro il radar posteriore.

Durante la costruzione di tutte le mappe si impone una probabilità di occupazione pari a 0 sotto il livello di terra e una probabilità pari a 0.5 al di sopra dei radar, per eliminare le false rilevazioni sistematiche notate durante la fase sperimentale. Queste operazioni non comportano perdita di informazioni in quanto non intervengono sull'area di interesse, l'unica che potrebbe contenere ostacoli potenzialmente pericolosi durante l'apertura delle portiere dell'auto.

Dai risultati ottenuti si osserva una buona corrispondenza tra l'ambiente reale e quello ricostruito grazie tramite i segnali radar, pur notando che per l'identificazione del target si ricava sempre un numero di punti abbastanza basso. Ciò potrebbe essere dovuto al fatto che, in ambito urbano, gli ostacoli indagati sperimentalmente non sono mai di materiale metallico, cosa che diminuisce la loro capacità di riflettere i segnali radar. Un altro potenziale problema è che a queste frequenze, ostacoli piatti (come ad esempio un muro) visti con angolo d'incidenza elevato potrebbero non generare riflessioni significative, per cui potrebbero non essere individuati. Allargando però il raggio dei punti individuati tramite la funzione *inflate* di Matlab, si ottengono volumi più pieni, che rispecchiano chiaramente il target reale.

Metodo 2

Il secondo approccio prevede invece di mantenere una soglia molto bassa (-100 dB) durante la rilevazione sperimentale, salvando nella matrice *TargetData* anche molti punti dovuti potenzialmente al rumore. In questo caso viene leggermente modificato il codice di acquisizione, in modo da memorizzare oltre alla coordinata di ogni punto rilevato anche la potenza del segnale corrispondente. All'interno del codice di mapping si associano proporzionalmente le intensità dei punti a delle probabilità comprese tra 0.5 e 1 utilizzando la funzione *normalize(potenze,'range',[0.5 1])* che assegna a valori di potenza una probabilità secondo la seguente relazione:

$$prob_i = 0.5 + \frac{potenze_i - min_{potenze}}{max_{potenze} - min_{potenze}} (1 - 0.5) \quad (5.3)$$

Queste probabilità, diverse per ogni punto, vengono poi utilizzate per aggiornare l'occupazione delle celle tramite la funzione *updateOccupancy*. Oltre alla OGM 3D si producono grafici raffiguranti tutti i punti acquisiti dai radar, e non solo quelli che alla fine dell'algoritmo risultano avere probabilità superiori a 0.65. Grazie a questa operazione è possibile verificare che i punti riferiti all'ostacolo non vengano effettivamente scartati. Si colora ogni punto con la sua probabilità di occupazione associata all'intensità dello stesso.

Si riportano in seguito i risultati ottenuti rielaborando con questo secondo approccio i dati raccolti dallo scenario 1.

Scenario 1

Muro distante 60 cm, alto 150 cm.

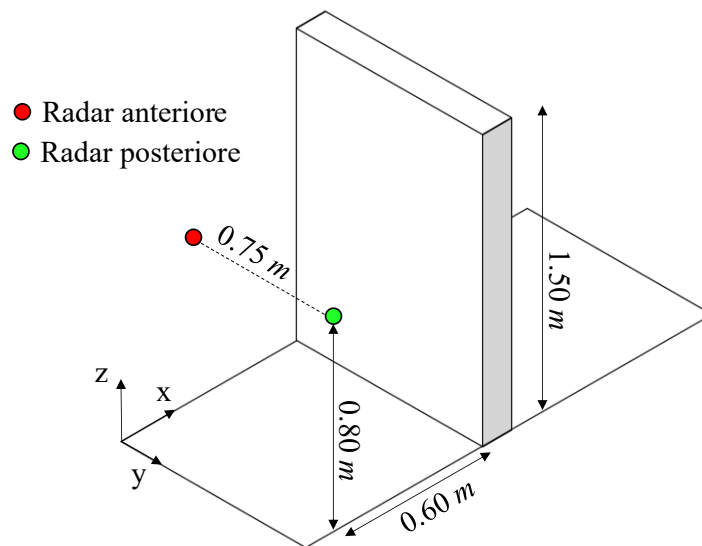


Figura 5.17 Rappresentazione grafica dello scenario n° 1

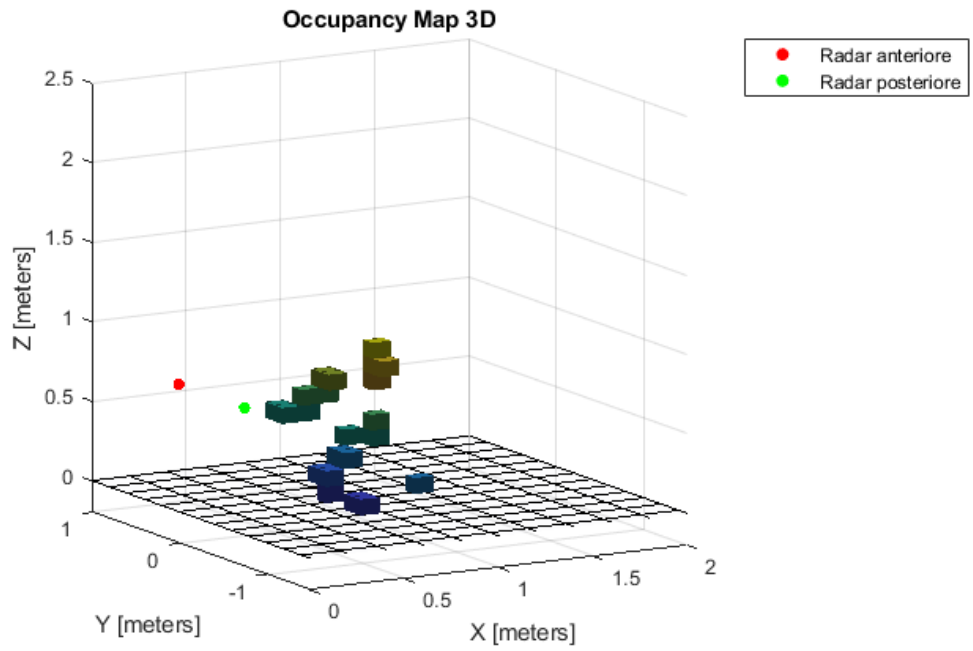


Figura 5.18 OGM 3D

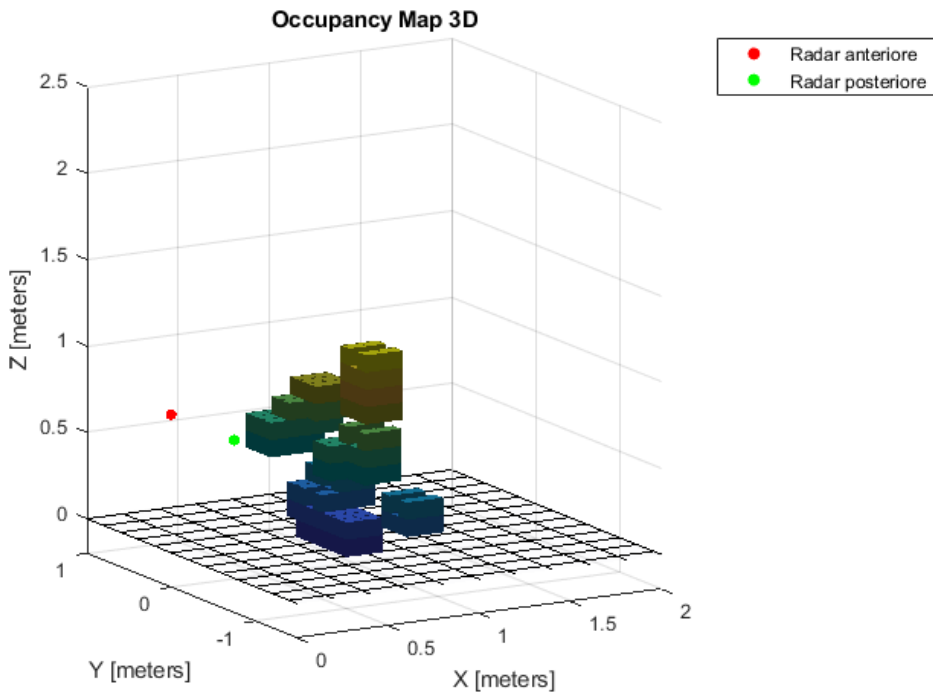


Figura 5.19 OGM 3D, inflate

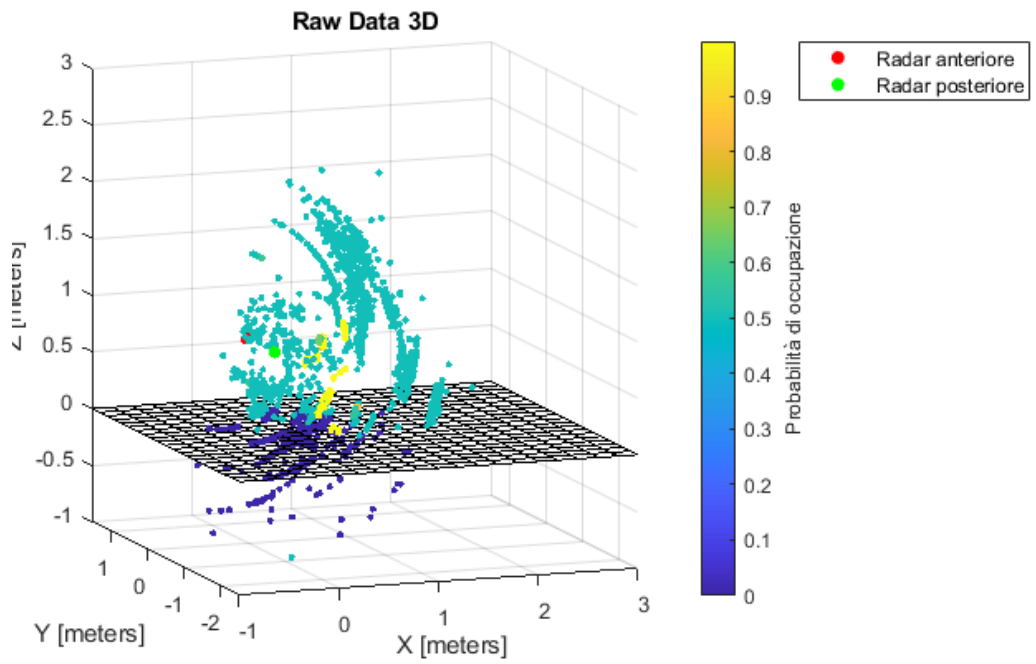


Figura 5.20 Raw Data 3D

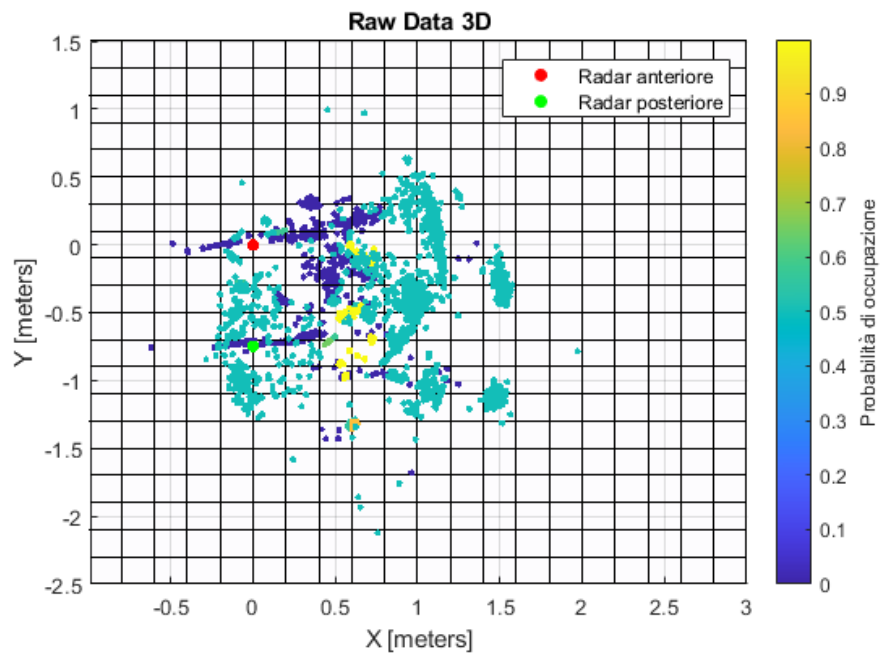


Figura 5.21 Raw Data 3D, piano (x, y)

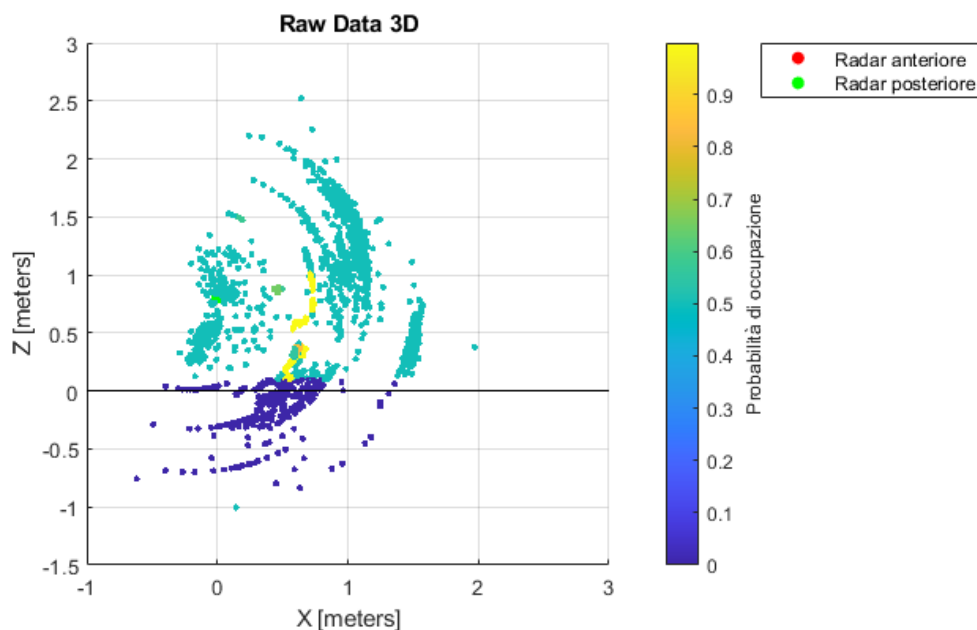


Figura 5.22 Raw Data 3D, piano (x, z). Secondo questo punto di vista il radar anteriore si trova dietro il radar posteriore.

Dai grafici Raw Data (5.20), (5.21) e (5.22) è possibile osservare la totalità dei punti raccolti dai radar. Si nota che le rilevazioni sono molto rumorose ma l’algoritmo di mapping riesce ad associare la probabilità corretta ad ogni punto, permettendo quindi di individuare chiaramente il profilo del muro presente di fianco all’automobile (punti di colore giallo, ovvero con probabilità di occupazione vicina a 1).

Confrontando le OGM 3D con i grafici Raw Data 3D si nota che la scarsa quantità di punti responsabili dell’individuazione del target non è dovuta ad una scrematura eccessiva; anche mantenendo la soglia di acquisizione bassa ed integrando probabilisticamente ogni punto raccolto in base alla potenza ricevuta le mappe sono del tutto simili a quelle derivate con il metodo 1. Ciò porta a supporre che la carenza di punti sia effettivamente dovuta al materiale del target, non metallico e quindi non pienamente riflettente. D’altra parte, osservando i valori di occupazione di ogni cella al termine dell’algoritmo, si nota che il metodo 2 porta a probabilità risultanti più differenziate tra loro; infatti, utilizzando una probabilità di aggiornamento legata alla potenza del segnale ricevuto (e quindi diversa per ogni punto), ci si aspetta una maggiore precisione nel calcolo della probabilità finale di ogni cella.

In generale tutti i risultati ottenuti sono conformi al reale posizionamento dei vari target nello spazio, si può quindi affermare che la tecnica di mapping utilizzata è efficace e permette una fedele riproduzione tridimensionale dell'ambiente.

Conclusioni

I modelli tridimensionali forniscono una rappresentazione volumetrica dello spazio che è fondamentale per un'ampia varietà di applicazioni nel campo automotive. La mappa è infatti una componente centrale di qualsiasi sistema autonomo, poiché su di essa si basano la pianificazione e l'esecuzione dell'azione. La maggior parte di queste applicazioni richiede una rappresentazione probabilistica dei dati rilevati, che consiste nella modellazione di aree libere, occupate e sconosciute. Le misurazioni raccolte dai sensori sono infatti afflitte da incertezza: oltre a errori sistematici potrebbero essere presenti anche rilevazioni apparentemente casuali dovute a riflessioni indesiderate o ostacoli dinamici. Quando l'obiettivo è creare un modello accurato dell'ambiente da tali misurazioni rumorose, l'incertezza deve essere necessariamente presa in considerazione in modo probabilistico. Più misurazioni incerte possono quindi essere fuse tra loro per ottenere una stima più robusta del vero stato dell'ambiente. Grazie alla fusione probabilistica dei dati è quindi possibile l'integrazione di misurazioni provenienti da più sensori che indagano contemporaneamente lo stesso ambiente; nel caso in questione due radar posti sulla fiancata di un'automobile.

Lo scopo principale della tesi è stato quello di implementare un algoritmo di mapping mediante segnali radar, che riproduca efficacemente l'ambiente 3D indagato. In particolare, è stata condotta una campagna sperimentale di misurazione in un ambiente realistico per verificare e convalidare l'efficacia del mapping. La configurazione sperimentale consiste nel posizionamento di due sensori sulle maniglie anteriore e posteriore di un'automobile, in modo da acquisire dati sull'ambiente circostante da due diverse angolazioni, per poi fondere le informazioni ottenute in un'unica mappa tridimensionale. Nel capitolo 3 sono state illustrate le caratteristiche generali del dispositivo utilizzato, il radar FMCW BGT60ATR24C di Infineon.

L'approccio di mappatura utilizzato in questa tesi consiste in una rappresentazione dello spazio tridimensionale basata su una struttura dati octree e utilizza la stima probabilistica dell'occupazione. La mappa ottenuta rappresenta esplicitamente non solo lo spazio occupato, ma

anche aree libere e sconosciute. Per convalidare il metodo di mappatura sono stati presentati nel capitolo 5 una serie di risultati sperimentali, ottenuti elaborando dati raccolti da un sensore reale posto in uno scenario di acquisizione reale. Utilizzando lo stesso codice di mapping sono stati considerati due diversi approcci alla costruzione della OGM. Il primo prevede una “scrematura” dei punti rumorosi durante la fase di acquisizione, utilizzando poi durante la creazione della mappa la stessa probabilità di aggiornamento per ogni punto ottenuto, supponendo che tutti i punti con intensità superiore alla soglia di acquisizione siano effettivamente target. La seconda modalità prevede invece il mantenimento in memoria di tutti i punti rilevati, compresi quelli potenzialmente causati dal rumore, tenendo una soglia molto bassa durante l’acquisizione. In questo caso vengono inoltre memorizzate le potenze ricevute per ogni punto acquisito, che verranno poi tradotte nelle probabilità di aggiornamento associate a ogni punto durante la creazione della OGM.

In entrambi i casi i risultati dimostrano che l’approccio scelto è in grado di aggiornare la rappresentazione probabilistica in modo efficiente e modellare i dati in modo coerente con la realtà. Nonostante ciò si nota sempre una carenza di punti volti all’individuazione del target, probabilmente dovuta al fatto che in ambiente urbano gli ostacoli indagati non sono in materiale metallico, e quindi non pienamente riflettente, specialmente per angoli d’incidenza elevati.

A partire dallo studio svolto in questa tesi, i prossimi sviluppi di ricerca potrebbero riguardare la possibilità di rilevare non solo un ambiente statico, ma anche ostacoli in movimento. Una volta creata una mappa che rilevi appropriatamente l’ambiente che circonda l’auto, sarebbe utile sviluppare un’applicazione automotive che, partendo dalla OGM, implementi ad esempio funzioni di bloccaggio della portiera oppure che avvisino l’utente nel caso in cui durante l’apertura si rilevi un ostacolo potenzialmente pericoloso.

A. Sensing GUI

La sensing GUI (*Graphical User Interface*) è una piattaforma che abilita la configurazione dei parametri e l’acquisizione dei dati dal sensore radar Infineon BGT60ATR24C. Dopo aver connesso tramite porta USB il sensore al PC è disponibile un’interfaccia grafica che permette l’interazione con l’utente attraverso diverse *tab* che visualizzano i segnali d’interesse. Situate sul lato sinistro della schermata principale della GUI sono invece presenti le schede predisposte alla configurazione dei chirp che comporranno il frame.

A.1 Configurazione del frame

A.1.1 Scheda global

La configurazione dello *shape set* avviene tramite la scheda “Global”.

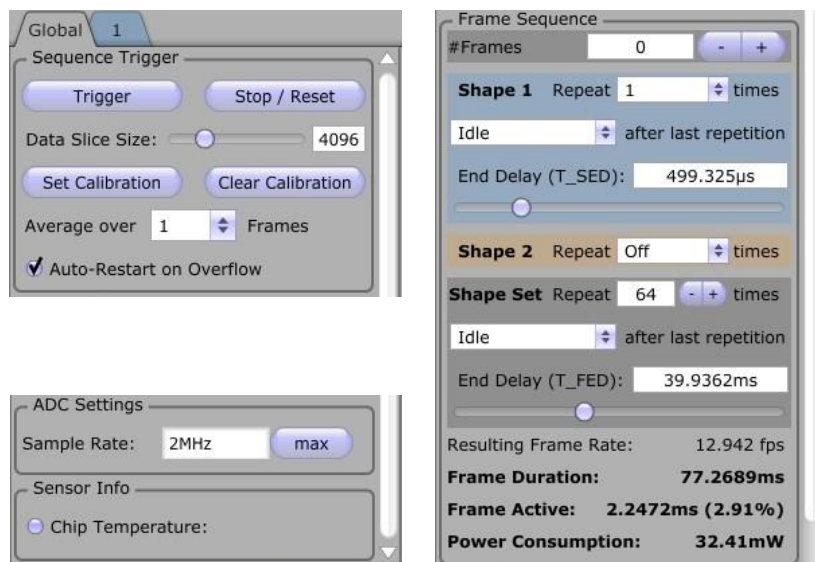


Figura A.1 Scheda Global

Per prima cosa sono presenti due comandi per iniziare e terminare l’acquisizione dei dati, il comando “*trigger*” e il comando “*stop/reset*”.

Per quanto riguarda il frame è possibile stabilire: il numero di *shape set* di cui esso sarà composto, le varie shape presenti nel singolo *shape set* (sono disponibili al massimo 4 shape differenti), ed il numero totale di frame trasmessi durante tutto il tempo di acquisizione (il valore zero significa ripetizioni infinite). Per ogni shape e per l'intero *shape set* è possibile scegliere anche l'*end delay*, ovvero il tempo trascorso dalla fine della trasmissione della forma corrente e l'inizio della trasmissione successiva. Infine sono visualizzati altri parametri del frame come la frequenza di ripetizione, la durata, il tempo di attività del chip durante il frame o il consumo energetico stimato per la configurazione scelta.

A.1.2 Schede di shape

Ciascuna forma attivata appare come una scheda di diverso colore. Selezionando una scheda di shape, è possibile configurare la forma corrispondente.

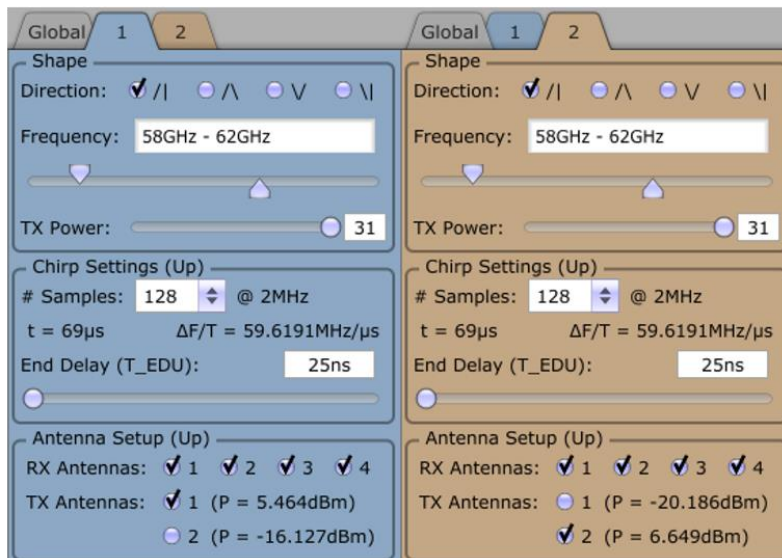


Figura A.2 Schede di shape

Innanzitutto è possibile scegliere la direzione del chirp: “up”, “up-down”, “down-up” o “down”. Ad esempio, selezionando “up” la frequenza del segnale crescerà linearmente nel tempo, mentre selezionando “down” decrescerà. La larghezza di banda del chirp può essere configurata con un campo di modifica testo oppure grazie a due cursori che indicano la frequenza minima e massima. Si può inoltre regolare la potenza del trasmettitore su una scala [0-31], dove 31 è la potenza massima e 0 è la potenza minima. Per ciascuna direzione di ciascuna forma è consentita infine la configurazione di numero di campioni acquisiti e antenne utilizzate per la trasmissione/ricezione della stessa.

A.2 Acquisizione dati

A.2.1 Timing tab

La scheda “Timing” mostra nel tempo l’andamento del frame scelto. In particolare sono riportate le durate di ogni fase di trasmissione e ricezione dei segnali.

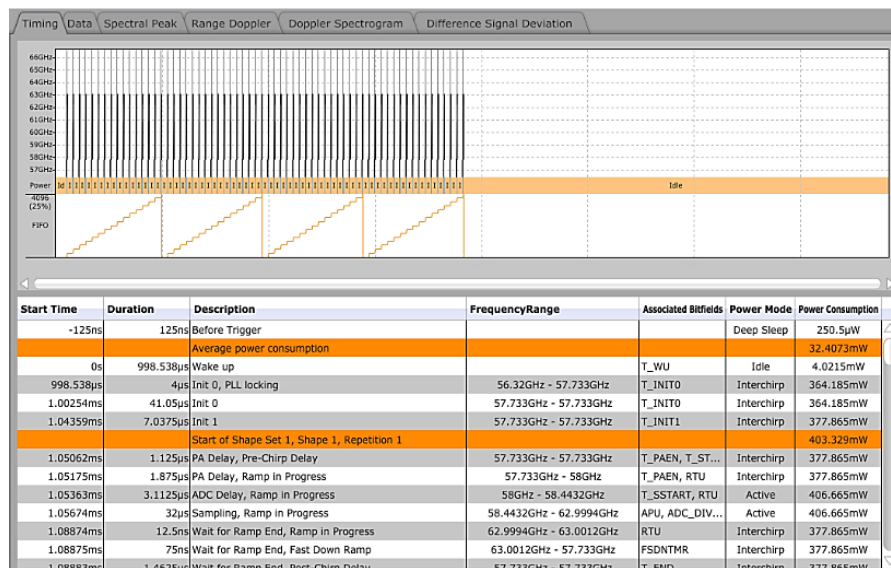


Figura A.3 Timing tab

A.2.2 Data tab

Nella parte superiore è mostrato il segnale ricevuto [mV] nel dominio del tempo [μs], distinguendo le shape che formano il frame: ogni riquadro ospita una shape appartenente al set configurato, set ripetuto N volte (con N definito dal parametro “*shape set repeat*”) fino a formare il frame. I chirp ricevuti dalle varie antenne sono distinti grazie a colori diversi per le diverse antenne RX abilitate: RX1 in verde, RX2 in rosso, RX3 in blu, RX4 in marrone. Nella parte inferiore è invece riportato lo spettro nel dominio delle frequenze: [kHz] sull’asse x e [dB] sull’asse y.

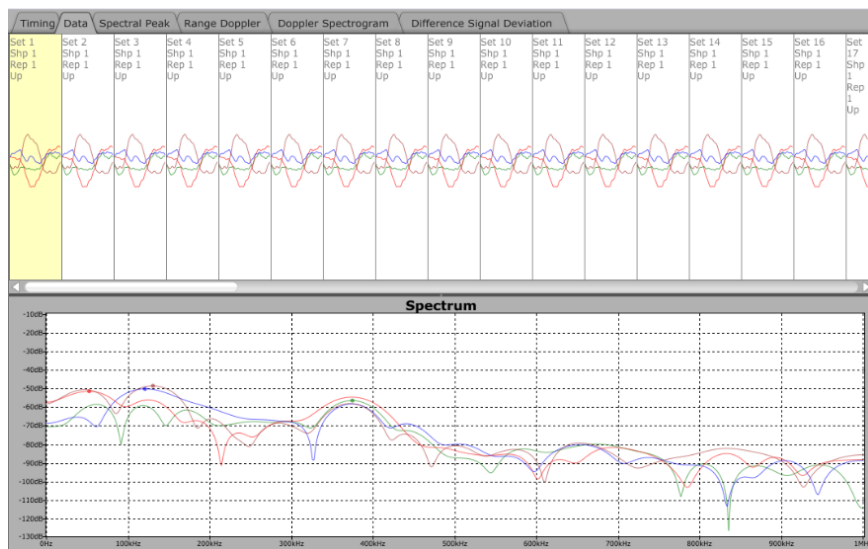


Figura A.4 Data tab

A.2.3 Range Doppler tab

Per visualizzare dati sulla Range Doppler tab è innanzitutto necessario scegliere il chirp da visualizzare dal menù che si apre cliccando il tasto destro del mouse. I grafici (uno per ogni antenna RX attiva) riportano sulle ascisse la velocità radiale del target e sulle ordinate la distanza del

target dal sensore. Se la velocità è positiva si ha un target in avvicinamento, se invece è negativa si ha un target in allontanamento. Il colore (da giallo a rosso) rappresenta la “forza” del target, in decibel Full-Scale (dBFS). Per distinguere i dati dal rumore di fondo, è possibile regolare una soglia facendo clic destro su uno dei grafici e selezionando “*threshold*” e quindi il valore desiderato. Impostando la soglia si visualizzano solo i target più “forti” di essa. Inoltre, il filtro MTI (*Moving Target Indicator*) può essere abilitato per rilevare solo i target in movimento e ignorare eventuali target fissi.

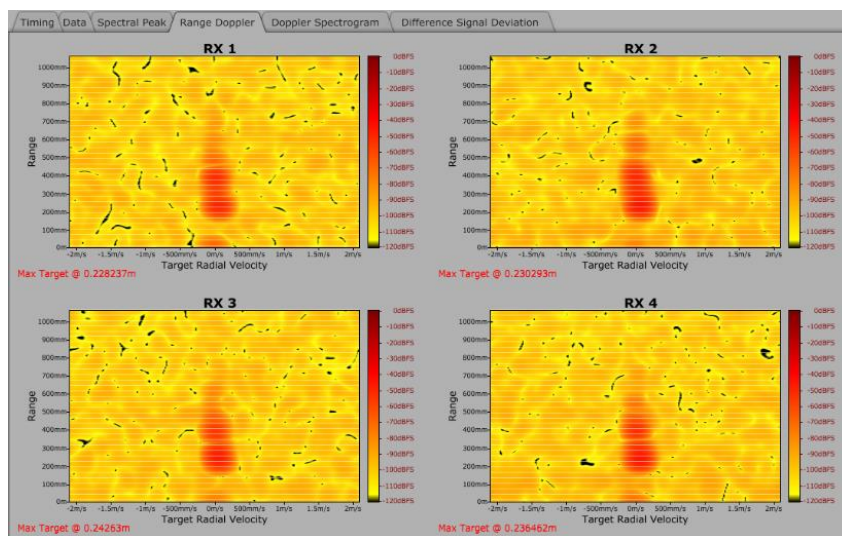


Figura A.5 Range Doppler tab

B. Codice Matlab

Occupation Grid-Mapping 3D

```
clc
clear all;

radar_height = 0.8; %altezza radar
shift = -0.75; %distanza tra le maniglie dell'auto

%creazione dell'occupation gridmap 3D
occ = 0.65; %probabilità >=0.65 per cella occupata
res = 10; %risoluzione, numero di celle per metro
%creazione di una mappa 3D vuota
map3D = occupancyMap3D(res, 'OccupiedThreshold', occ);

%impostazione della mappa a probabilità iniziale 0.5
[xStart, yStart, zStart] = meshgrid(-2:0.1:2, -2:0.1:2, -1:0.1:3);
xyzStart = [xStart(:) yStart(:) zStart(:)];
occValStart = 0.5;
setOccupancy(map3D, xyzStart, occValStart);

load 'TargetData_a.mat'; %dati RADAR ANTERIORE
load 'POTENZE_a.mat'; %potenze dati RADAR ANTERIORE
%trasformazione da potenze a probabilità [0.5 1]
power1 = normalize(potenze, 'range', [0.5 1]);

range_a = [];
azimuth_a = [];
elevation_a = [];
pt_a = [];
x_a = [];
y_a = [];
z_a = [];

%per ogni frame i si estraggono i punti rilevati e si
aggiorna probabilità di occupazione
for i = 1:length(TargetData)
    rilevazione_a = TargetData{1,i};
    range_a = [rilevazione_a(:,1)];
    azimuth_a = [rilevazione_a(:,2)];
    elevation_a = [rilevazione_a(:,3)];
end
```

```

    %coordinate cartesiane target al frame i
    [Xa,Ya,Za] =
sph2cart(deg2rad(azimuth_a),deg2rad(elevation_a),range_a
);
    points_a = [Xa, Ya, Za+radar_height];

    prob=[]; %terzo argomento di updateOccupancy
    for j = 1:length(points_a)
        prob = [prob; power1(j)];
    end

    pt_a = [pt_a; points_a];
    x_a = [x_a; Xa];
    y_a = [y_a; Ya];
    z_a = [z_a; Za+radar_height];
    %aggiornamento probabilità punti rilevati al frame i
    updateOccupancy(map3D,points_a,prob);
end

load 'TargetData_p.mat'; %dati RADAR POSTERIORE
load 'POTENZE_p.mat'; %potenze dati RADAR POSTERIORE
%trasformazione da potenze a probabilità [0.5 1]
power2 = normalize(potenze,'range',[0.5 1]);

range_p = [];
azimuth_p = [];
elevation_p = [];
pt_p = [];
x_p = [];
y_p = [];
z_p = [];

%per ogni frame i si estraggono i punti rilevati e si
aggiorna probabilità di occupazione
for i = 1:length(TargetData)
    rilevazione_p = TargetData{1,i};
    range_p = [rilevazione_p(:,1)];
    azimuth_p = [rilevazione_p(:,2)];
    elevation_p = [rilevazione_p(:,3)];
    %coordinate cartesiane punti rilevati al frame i
    [Xp,Yp,Zp] =
sph2cart(deg2rad(azimuth_p),deg2rad(elevation_p),range_p
);
    points_p = [Xp, Yp+shift, Zp+radar_height-0.1];

```

```

prob=[]; %terzo argomento di updateOccupancy
for j = 1:length(points_p)
    prob = [prob; power2(j)];
end

pt_p = [pt_p; points_p];
x_p = [x_p; Xp];
y_p = [y_p; Yp+shift];
z_p = [z_p; Zp+radar_height-0.1];
%aggiornamento probabilità punti rilevati al frame i
updateOccupancy(map3D,points_p,prob);
end

%si impone occupazione a 0 sotto il livello di terra
[xGround,yGround,zGround] = meshgrid(-2:0.05:2,-
2:0.05:2,-1:0.05:0);
xyzGround = [xGround(:) yGround(:) zGround(:)];
occValGround = 0;
setOccupancy(map3D,xyzGround,occValGround);

%si impone occupazione a 0 sotto il livello dei radar
[xUp,yUp,zUp] = meshgrid(-2:0.05:2,-
2:0.05:2,radar_height:0.05:4);
xyzUp = [xUp(:) yUp(:) zUp(:)];
occValUp = 0;
setOccupancy(map3D,xyzUp,occValUp);

occVal_a = [];
occVal_a = getOccupancy(map3D,pt_a) %valori di
occupazione dei punti dal radar anteriore
occVal_p = [];
occVal_p = getOccupancy(map3D,pt_p) %valori di
occupazione dei punti dal radar posteriore

%allargamento del raggio dei punti plottati
inflationRadius = 0.05;
inflate(map3D, inflationRadius);

% numRays = 20; %raggi visivi che partendo dal sensore
incontrano un ostacolo
% xRay = linspace(0,2,numRays);
% yRay = linspace(-1,1,numRays);
% directions = [xRay; yRay; zeros(1,numRays)]'; %-
0.5*ones(1,numRays)]';
% sensorPose_a = [0 0 radar_height 1 0 0 0];
%ANTERIORE

```

```

% sensorPose_p = [0 shift radar_height 1 0 0 0];
%POSTERIORE
% maxrange = 5;
% [intersectionPts,isOccupied] =
rayIntersection(map3D,sensorPose_a,directions,maxrange);
% [intersectionPts1,isOccupied1] =
rayIntersection(map3D,sensorPose_p,directions,maxrange);

% mappa 3D delle celle occupate da terget, colore in base
all'altezza del target
figure (1)
show(map3D)
title('Occupancy Map 3D');
grid
xlim([0 2]);
ylim([-1.5 1]);
zlim([-0.2 2.5]);
hold on
% for i = 1:numRays %plot raggi visivi che partendo dal
sensore incontrano un ostacolo
%
%plotTransforms(sensorPose(1:3),sensorPose(4:end));
%   if isOccupied(i) == 1
%       plot3([sensorPose_a(1),intersectionPts(i,1)],...
%           [sensorPose_a(2),intersectionPts(i,2)],...
%           [sensorPose_a(3),intersectionPts(i,3)],'*r')
%   end
%   if isOccupied1(i) == 1
%       plot3([sensorPose_p(1),intersectionPts1(i,1)],...
%           [sensorPose_p(2),intersectionPts1(i,2)],...
%           [sensorPose_p(3),intersectionPts1(i,3)],'-*g')
%   end
% end
hold on

radar_a = scatter3(0,0,radar_height,30,'r','filled');
%punto rosso = radar anteriore
hold on

radar_p=scatter3(0,shift,radar_height,30,'g','filled');
%punto verde = radar posteriore
hold on

[a,b] = meshgrid(-2:0.2:2,-2:0.2:2);
c = zeros(21,21);

```

```

surf(a,b,c, 'FaceAlpha',0.01); %plot superficie terreno
legend([radar_a radar_p], {'Radar anteriore', 'Radar
posteriore'});
hold off

%creazione di una matrice 2D con all'interno i valori di
occupazione alla loro coordinata xy
xy_m = [y_a+2 x_a+1; y_p+2 x_p+1];
%matrice coordinate xy dei target in m
mat_dm = round(xy_m*10);
%matrice coordinate xy dei target in dm
A = 0.5*ones(30); %matrice ambiente 3x3 m, a priori
ogni cella ha occupazione 0.5
occVal = [occVal_a; occVal_p]; %vettore valori di
occupazione per le celle con target

%inserimento dei valori di occupazione nelle celle con
target
for i = 1:length(mat_dm)
    a = mat_dm(i,1);
    b = mat_dm(i,2);
    A(a,b) = occVal(i);
end

%creazione di una matrice 2D con all'interno i valori di
occupazione alla loro coordinata xz
xz_m = [z_a+1 x_a+1; z_p+1 x_p+1];
%matrice coordinate xy dei target in m
mat_dm2 = round(xz_m*10);
%matrice coordinate xy dei target in dm
B = 0.5*ones(30);
%matrice ambiente 3x3 m, a priori ogni cella ha
occupazione 0.5

%inserimento dei valori di occupazione nelle celle che
contengono target
for i = 1:length(mat_dm2)
    a = mat_dm2(i,1);
    b = mat_dm2(i,2);
    B(a,b) = occVal(i);
end

B(10,:) = 0; %ground con occupazione 0

```

```

%mappa 2D, colorazione in base al valore di occupazione
delle celle, piano(x,y)
figure(2)
s = pcolor(A);
xticklabels({'-0.5', '0', '1.5', '1', '1.5', '2'});
yticklabels({'-1.5', '-1', '-0.5', '0', '0.5', '1'});
xlabel('X [meters]');
ylabel('Y [meters]');
title('Occupancy Map 2D');
colormap winter(10);
s.FaceColor = 'interp';
h = colorbar;
set(get(h, 'label'), 'string', 'Probabilità di
occupazione');
hold on

r_a = scatter(10,20,30, 'r', 'filled');
%punto rosso = radar anteriore
hold on

r_p = scatter(10,20+shift*10,30, 'g', 'filled');
%punto verde = radar posteriore
legend([r_a r_p], {'Radar anteriore', 'Radar
posteriore'});
hold off

%mappa 2D, colorazione in base al valore di occupazione
delle celle, piano(x,z)
figure(3)
s = pcolor(B);
xticklabels({'-0.5', '0', '0.5', '1', '1.5', '2'});
yticklabels({'-0.5', '0', '0.5', '1', '1.5', '2'});
xlabel('X [meters]');
ylabel('Z [meters]');
title('Occupancy Map 2D');
colormap winter(10);
s.FaceColor = 'interp';
h = colorbar;
set(get(h, 'label'), 'string', 'Probabilità di
occupazione');
hold on

r_a = scatter(10,10+radar_height*10,30, 'r', 'filled');
%punto rosso = radar anteriore
hold on

```



```

r_p = scatter(10,10+radar_height*10,30,'g','filled');
%punto verde = radar posteriore
legend([r_a r_p],{'Radar anteriore','Radar
posteriore'});
hold off

%plot dei grafici Raw Data, colore in base alla
probabilità di occupazione
figure(4)
scatter3(x_a,y_a,z_a,10,occVal_a,'filled');
title('Raw Data 3D');
%view(2)
hold on

scatter3(x_p,y_p,z_p,10,occVal_p,'filled');
h = colorbar;
set(get(h,'label'),'string','Probabilità di
occupazione');
xlabel('X [meters]');
ylabel('Y [meters]');
zlabel('Z [meters]');
hold on

radar_a = scatter3(0,0,radar_height,30,'r','filled');
%punto rosso = radar anteriore
hold on

radar_p =
scatter3(0,shift,radar_height,30,'g','filled');
%punto verde = radar posteriore
hold on

%plot superficie terreno
[a,b] = meshgrid(-1:0.2:3,-2.5:0.2:1.5);
c = zeros(21,21);
surf(a,b,c,'FaceAlpha',0.01);
legend([radar_a radar_p],{'Radar anteriore','Radar
posteriore'});
hold off

```


Lista delle figure

1.1 Schema a blocchi di un generico sistema radar.....	3
1.2 Segnale 'chirp' nel tempo.....	6
1.3 Schema a blocchi di un sistema radar FMCW	7
1.4 TX chirp e RX chirp.....	8
1.5 Angolo di arrivo del segnale ricevuto	11
2.1 Confronto tra OGM ed AGM [8]	21
3.1 Infineon radar BGT60ATR24C.....	24
3.2 Rappresentazione dei target rilevati: Range, Azimuth ed Elevation.....	27
3.3 Configurazione frame (immagine da Sensing GUI).....	28
3.4 Configurazione shape set (immagine da Sensing GUI)	29
3.5 Matrice “ <i>RawData</i> ” a sinistra e matrice “ <i>TargetData</i> ” a destra ..	30
4.1 Rappresentazione grafica degli esperimenti in scenario reale.....	31
4.2 Scenario controllato per esperimenti preliminari	32
4.3 Scenario controllato per esperimenti preliminari	32
4.4 Sistema di riferimento	33
4.5 Scenario n° 1	33
4.6 Scenario n° 1, risultati	34
4.7 Scenario n° 2	35
4.8 Scenario n° 2, risultati	35
4.9 Scenario n° 3	36
4.10 Scenario n° 3, risultati	36
4.11 Esperimenti in uno scenario statico reale	37
4.12 Scenario reale n° 1.....	39
4.13 Scenario reale n° 2.....	40
4.14 Scenario reale n° 3.....	41
4.15 Scenario reale n° 4, da portiera anteriore	42

4.16 Scenario reale n° 5, da portiera anteriore	43
5.1 Octree data structure [12]	45
5.2 Rappresentazione grafica dello scenario n° 1.....	48
5.3 OGM 3D.....	49
5.4 OGM 3D, inflate.....	49
5.5 OGM 2D, piano (x, y)	50
5.6 OGM 2D, piano (x, z). Secondo questo punto di vista il radar anteriore si trova dietro il radar posteriore	50
5.7 Rappresentazione grafica dello scenario n° 2.....	51
5.8 OGM 3D.....	51
5.9 OGM 3D, inflate.....	52
5.10 OGM 2D, piano (x, y)	52
5.11 OGM 2D, piano (x, z). Secondo questo punto di vista il radar anteriore si trova dietro il radar posteriore	53
5.12 Rappresentazione grafica dello scenario n° 3.....	53
5.13 OGM 3D.....	54
5.14 OGM 3D, inflate.....	54
5.15 OGM 2D, piano (x, y)	55
5.16 OGM 2D, piano (x, z). Secondo questo punto di vista il radar anteriore si trova dietro il radar posteriore	55
5.17 Rappresentazione grafica dello scenario n° 1.....	57
5.18 OGM 3D.....	58
5.19 OGM 3D, inflate.....	58
5.20 Raw Data 3D	59
5.21 Raw Data 3D, piano (x, y).....	59
5.22 Raw Data 3D, piano (x, z). Secondo questo punto di vista il radar anteriore si trova dietro il radar posteriore	60
A.1 Scheda Global.....	64

A.2 Schede di shape	65
A.3 Timing tab	66
A.4 Data tab.....	67
A.5 Range Doppler tab.....	68

Bibliografia

- [1] Texas Instruments. “*The fundamentals of millimeter wave sensors*”. URL: <https://www.ti.com/lit/pdf/spyy005>, July 2020.
- [2] Kingsley S. & Quegan S. (1999). “*Understanding radar systems*” (Vol. 2). SciTech Publishing, pp. 11-15.
- [3] Texas Instruments. “*Introduction to mmwave Sensing: FMCW Radars*”. URL: https://training.ti.com/sites/default/files/docs/mmwaveSensing-FMCW-offlineviewing_0.pdf
- [4] Texas Instruments. “MIMO Radar”. *Application Report SWRA554A*, May 2017–Revised July 2018. URL: <https://www.ti.com/lit/pdf/swra554>
- [5] Zhou T., Yang M., Jiang K., Wong H. & Yang D. (2020). “MMW Radar-Based Technologies in Autonomous Driving: A Review”. *Sensors*, 20(24), 7283.
- [6] Sless L., El Shlomo B., Cohen G., & Oron S. (2019). “Road scene understanding by occupancy grid learning from sparse radar clusters using semantic segmentation”. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops* (pp. 0-0).
- [7] Thrun S. (2003). “Learning occupancy grid maps with forward sensor models”. *Autonomous robots*, 15(2), 111-127.
- [8] Werber K., Rapp M., Klappstein J., Hahn M., Dickmann J., Dietmayer K., & Waldschmidt C. (2015, April). “Automotive radar gridmap representations”. In *2015 IEEE MTT-S International Conference on Microwaves for Intelligent Mobility (ICMIM)* (pp. 1-4). IEEE.

- [9] Bagnell D., "Occupancy Maps", *Statistical Techniques in Robotics* (16-831, F10) Lecture#06 (Thursday September 11).
- [10] Dia R., Mottin J., Rakotovao T., Puschini D., & Leseq S. (Jul 2017). "Evaluation of Occupancy Grid Resolution through a Novel Approach for Inverse Sensor Modeling". *IFAC World Congress*, Toulouse, France. cea-01571087.
- [11] Slutsky M., & Dobkin D. (2019, June). "Dual inverse sensor model for radar occupancy grids". In 2019 *IEEE Intelligent Vehicles Symposium (IV)* (pp. 1760-1767). IEEE.
- [12] Hornung A., Wurm K. M., Bennewitz M., Stachniss C. & Burgard W. (7 February 2013). "OctoMap: an efficient probabilistic 3D mapping framework based on octrees". *Springer Science, Business Media New York* 2013.
- [13] Guerra A., Guidi F., Dall'Ara J., & Dardari D. (2018). "Occupancy Grid Mapping for Personal Radar Applications". In 2018 *IEEE Statistical Signal Processing Workshop (SSP)*
- [14] Guidi F., Guerra A. and Dardari, D. "Personal Mobile Radars with Millimeter-Wave Massive Arrays for Indoor Mapping," in *IEEE Transactions on Mobile Computing*, vol. 15, no. 6, pp. 1471-1484, 1 June 2016, doi: 10.1109/TMC.2015.2467373.
- [15] Guidi F., Mariani A., Guerra A., Dardari D., Clemente A. and D'Errico R., "Indoor Environment-Adaptive Mapping With Beamsteering Massive Arrays," in *IEEE Transactions on Vehicular Technology*, vol. 67, no. 10, pp. 10139-10143, Oct. 2018, doi: 10.1109/TVT.2018.2853657.
- [16] Guidi F., Guerra A., Dardari D., Clemente A. and D'Errico R., "Joint Energy Detection and Massive Array Design for Localization and Mapping," in *IEEE Transactions on Wireless Communications*, vol. 16, no. 3, pp. 1359-1371, March 2017, doi: 10.1109/TWC.2016.2627032.

- [17] Guerra A., Guidi F., Ahtaryieva L., Decarli N. and Dardari D., "Crowd-based personal radars for indoor mapping using UWB measurements," 2016 IEEE *International Conference on Ubiquitous Wireless Broadband (ICUWB)*, 2016, pp. 1-4, doi: 10.1109/ICUWB.2016.7790437.
- [18] Pasolini G., Guerra A., Guidi F., Decarli N., and Dardari D. "Crowd-Based Cognitive Perception of the Physical World: Towards the Internet of Senses. *Sensors*". 2020; 20(9):2437. URL: <https://doi.org/10.3390/s20092437>

Ringraziamenti

*Ringrazio innanzitutto il professor Davide Dardari,
per avermi accompagnata con grande disponibilità
durante tutta la stesura di questa tesi,
e per i suoi preziosi insegnamenti.*

*Ringrazio il professor Gianni Pasolini e la dottoressa Anna Guerra,
per avermi seguita lungo questo percorso,
e per tutti i loro consigli.*

*Infine ringrazio di cuore la mia famiglia,
per avermi supportata in ogni momento,
standomi vicino e facendo tutto il possibile per aiutarmi,
come sempre.*