

# RealSense Evaluator: valutazione delle performance di Intel RealSense T265 tracking camera in diversi contesti

Lorenzo De Luca

Matricola: 7037316

[lorenzo.deluca4@stud.unifi.it](mailto:lorenzo.deluca4@stud.unifi.it)

Francesco Gradi

Matricola: 7017190

[francesco.gradi@stud.unifi.it](mailto:francesco.gradi@stud.unifi.it)

Federico Màlato

Matricola: 7017325

[federico.malato@stud.unifi.it](mailto:federico.malato@stud.unifi.it)

## Abstract

*Il progetto che abbiamo svolto ha riguardato la valutazione di una tracking camera Intel RealSense T265 fornитaci dall'Università di Firenze. È stato utilizzato un Raspberry Pi 4 per andare a estrapolare i dati della fotocamera nei vari contesti operativi mobili della telecamera: in particolare il focus del progetto ha riguardato gli esperimenti di rilevazione della precisione di tracciamento in vari contesti di diversa natura, indoor e outdoor, con differente scale di grandezza e con diverse velocità di acquisizione. Abbiamo cercato di mettere in difficoltà la telecamera utilizzandola in spazi con sfondo uniforme, occludendo una fisheye, aggiungendo riferimenti particolari e modificando la velocità della stessa repentinamente. I risultati sono stati analizzati in seguito mediante software Python, sono stati inseriti grafici, valutati parametri di confidenza e sono stati effettuati confronti con altri dispositivi tecnologicamente differenti. Infine, nelle conclusioni, abbiamo tirato le somme e cercato di definire sistematicamente alcuni consigli utili per un uso futuro della camera.*

## 1. Introduzione

Lo scopo di questo progetto è quello di testare le funzionalità della videocamera di profondità Intel RealSense T265 in modo tale da poterne quantificare il grado di affidabilità nei vari contesti in cui la videocamera potrebbe essere inserita in un progetto futuro. L'analisi dei risultati ha mostrato che, nonostante alcuni errori dovuti ad approssimazioni e ad un environment di testing limitato, la telecamera funziona bene sia in ambienti indoor che outdoor. Particolarmente interessante è stata la scoperta di un "intervallo di calibrazione" necessario all'inizio di ogni utilizzo per poter ottenere risultati generalmente migliori.

## 2. Materiali e Metodi

Per gli esperimenti condotti abbiamo avuto bisogno sia di componenti hardware, sia di alcuni software. In questa sezione introduciamo brevemente ciascun componente in modo tale da fornire una panoramica più precisa delle nostre possibilità tecniche.

### 2.1. Componenti Hardware

#### 2.1.1 Intel RealSense T265

Intel RealSense T265 è una tracking camera presente sul mercato dal 2019. La telecamera è composta da un modulo di tracking affiancato da una serie di sensori. Le principali componenti sono:

- **IMU:** è una classe di sensori che comprendono un accelerometri e un giroscopi, disposti su tre assi, tramite cui è possibile ricavare i valori di accelerazioni e rotazioni in un sistema di coordinate ortonormale XYZ. In particolare, la telecamera fa uso del modello BMI055, che comprende su un unico sensore sia un accelerometro che un giroscopio con frequenze di campionamento (rispettivamente) a 62,5 Hz e 200 Hz.
- **VPU:** il Visual Processing Unit è ciò che rende una telecamera "intelligente": tramite questo componente è infatti possibile effettuare calcoli che permettono, ad esempio, di estrarre sul momento un maggior quantitativo di dati a partire dall'output dei sensori, oppure di ottenere risultati di algoritmi di visione (come gli algoritmi SLAM) in real time. La telecamera monta un Intel Movidius Myriad 2.
- **FishEye lens sensor:** la RealSense T265 è equipaggiata con due telecamere FishEye che permettono di effettuare una ricostruzione dello spazio circostante fedele e con un angolo di visione vasto. Le due telecamere funzionano a 30FPS e forniscono delle immagini monocromatiche.

Maggiori informazioni sono disponibili sul Datasheet ufficiale [1] della telecamera.

### 2.1.2 Raspberry Pi

Il Raspberry Pi è uno strumento naturalmente adatto ad effettuare prove che necessitino di grande mobilità: le sue dimensioni ridotte e la potenza di calcolo lo rendono il candidato perfetto per testare la videocamera in un grande numero di situazioni [2]. In particolare, per i nostri esperimenti abbiamo utilizzato un Raspberry Pi 4, in quanto decisamente più potente del modello precedente. Inoltre, abbiamo fatto qualche test anche con un Raspberry Pi 3 Model B così da poter delineare meglio i limiti del nostro setup. Il sistema operativo scelto è stato Raspbian. Entrambi i Raspberry sono stati equipaggiati con uno schermo touchscreen da 3.5" già in nostro possesso, così da poter garantire un controllo più agile durante le prove, e sono stati alimentati tramite un powerbank da 14000 mAh, 2.1 A di corrente in uscita e 5 V di tensione.

## 2.2. Componenti Software

### 2.2.1 Intel RealSense SDK

Intel mette a disposizione degli sviluppatori la RealSense SDK [3] per semplificare l'utilizzo, l'acquisizione e l'interpretazione dei dati della telecamera. L'utilizzo della SDK ha facilitato molto l'approccio iniziale alla videocamera e si è rivelato utile anche durante gli esperimenti, per comprenderne le falte e progettare possibili migliorie. Oltre all'interfaccia grafica immediata, è infatti possibile visualizzare i dati estratti dalla videocamera in real time, tra cui l'orientazione, il tracking del tracciato percorso dall'ultimo avvio e i valori numerici corrispondenti ai movimenti, tra cui posizione, velocità, accelerazione, angoli di yaw, pitch e roll e alcuni confidence scores.

Seguendo la guida ufficiale reperibile su GitHub, aggiornata al solo Raspberry Pi 3, ma compatibile anche con il 4, è stato compilato e installato l'SDK su Raspbian [4]. Tale processo ha richiesto, approssimativamente, 2-3 ore su Raspberry Pi 4 e 5-6 su Raspberry Pi 3B+.

### 2.2.2 Python

Dovendo lavorare principalmente sui dati, abbiamo deciso di utilizzare un linguaggio di programmazione che avesse tra i suoi punti di forza la gestione efficiente e il processing dei dati. A causa di alcune librerie sviluppate appositamente per questo scopo (tra cui numpy e matplotlib) la nostra scelta è ricaduta su Python

3.7. Un altro vantaggio di utilizzare questo linguaggio risiede nel fatto che le librerie proprietarie di Intel per il controllo e le acquisizioni della videocamera, pur essendo state scritte nativamente in C++, sono state wrappate in Python e, di conseguenza, abbiamo potuto accedervi direttamente per mezzo dello wrapper pyrealsense2 [5]. Sebbene utilizzare la libreria C++ nativa avrebbe migliorato le performance di acquisizione, sarebbe stata un'inutile complicazione del programma a fronte di un miglioramento risibile. Abbiamo dunque ritenuto sufficiente l'utilizzo dei wrappers ai nostri fini.

## 3. Setup

In questa breve sezione si vuole dare una panoramica di ciò che è stato fatto e degli espedienti per eseguire i test nella miglior condizione possibile. Inoltre, la sottosezione "Processing dei dati" fornisce uno scorcio del codice utilizzato per estrarre dati e costruire i grafici che verranno discussi nella sezione "Risultati".

### 3.1. Apparato di Visione

La telecamera è stata collegata tramite USB 3.0 al Raspberry Pi 4 (o USB 2.0 Raspberry Pi 3) e l'apparato è stato interamente alimentato dal powerbank. Date le grandi dimensioni di quest'ultimo rispetto al resto, è stato possibile utilizzarlo come "base" su cui poggiare gli altri moduli fissandoli con dello spago, grazie a un case ad hoc della camera Realsense. I cavi sono stati assicurati in modo tale da non risultare d'intralcio durante gli esperimenti.

### 3.2. Interfaccia Grafica

Come specificato nella sezione "Materiali e metodi", avevamo a disposizione uno schermo LCD touchscreen da 3.5" [6]. Abbiamo dunque pensato di sfruttare questa possibilità collegando lo schermo al Raspberry (tramite il connettore GPIO nativo) e sviluppando una piccola interfaccia grafica in Python (con PyQt5) che ci permettesse di avviare automaticamente gli script e le funzioni necessari per effettuare i test e salvarne i dati derivanti. L'interfaccia si compone di una serie di pulsanti corrispondenti ai test effettuabili e un timer per tenere traccia del tempo necessario al completamento del test.

### 3.3. Processing dei Dati

Il processing e il plotting dei dati è stato effettuato per mezzo delle librerie numpy e matplotlib. In particolare, i dati derivanti dagli esperimenti vengono salvati in formato .csv in real time durante lo svolgimento di un qualunque test e vengono recuperati per mezzo

di funzioni standard di Python. Siccome diversi esperimenti hanno avuto diversi scopi e hanno dunque richiesto dati differenti, abbiamo predisposto uno script per ciascuna tipologia di test in modo da non trattare i dati in modo confusionario e da avere un processing più specifico e funzionale possibile. Per i nostri test raccogliamo i valori delle posizioni rilevate dalla telecamera scisse nelle sue componenti x,y e z, così come le velocità e le accelerazioni. Inoltre, raccogliamo i dati derivanti dal giroscopio sulle rotazioni che la telecamera effettua rispetto al suo sistema di riferimento. Per ottenerli, abbiamo predisposto quattro tipi di script:

- Il primo tipo è stato utilizzato per ottenere il grafico dei percorsi effettuati e abbiamo previsto la possibilità di inserire anche una visualizzazione della velocità per mezzo di una colormap normalizzata sui valori di riferimento dell'esperimento. In questi script si ha una visualizzazione 3D del percorso unita alle visualizzazioni “a volo d'uccello” sui tre piani XY, XZ e YZ;
- Il secondo tipo di script di plotting si è focalizzato sul plotting delle accelerazioni e delle velocità da un punto di vista più telemetrico, attraverso grafici 2D più classici come “spazio-tempo” e “velocità-tempo” sui tre assi, in modo da poter ottenere un riscontro fisico di quanto testato;
- Il terzo tipo è stato molto utile per visualizzare i dati sulle rotazioni. In particolare questi script fanno uso di visualizzazioni 2D per visualizzare le rotazioni che avvengono lungo le tre componenti cartesiane;
- Il quarto ed ultimo script ci permette di visualizzare un istogramma degli score di confidenza ottenuti da un certo esperimento, basandoci sui dati della telecamera. In questo modo possiamo effettivamente verificare e soppesare la correttezza dei dati.

## 4. Esperimenti

Dato che la videocamera si presta bene ad essere utilizzata sia in un ambiente chiuso che in un ambiente aperto, abbiamo effettuato esperimenti in tutte e due le condizioni. In particolare sia al chiuso che all'aperto abbiamo effettuato cinque tipi di esperimenti, classificati in base allo spazio richiesto. In questa sezione vengono descritti il funzionamento e i setup utilizzati, mentre nella sezione “Risultati” analizzeremo i dati in maniera da poterne trarre alcune conclusioni. Per ciascun tipo di esperimento, inoltre, verranno indicati i nomi delle cartelle del progetto [7] che contengono i dati.

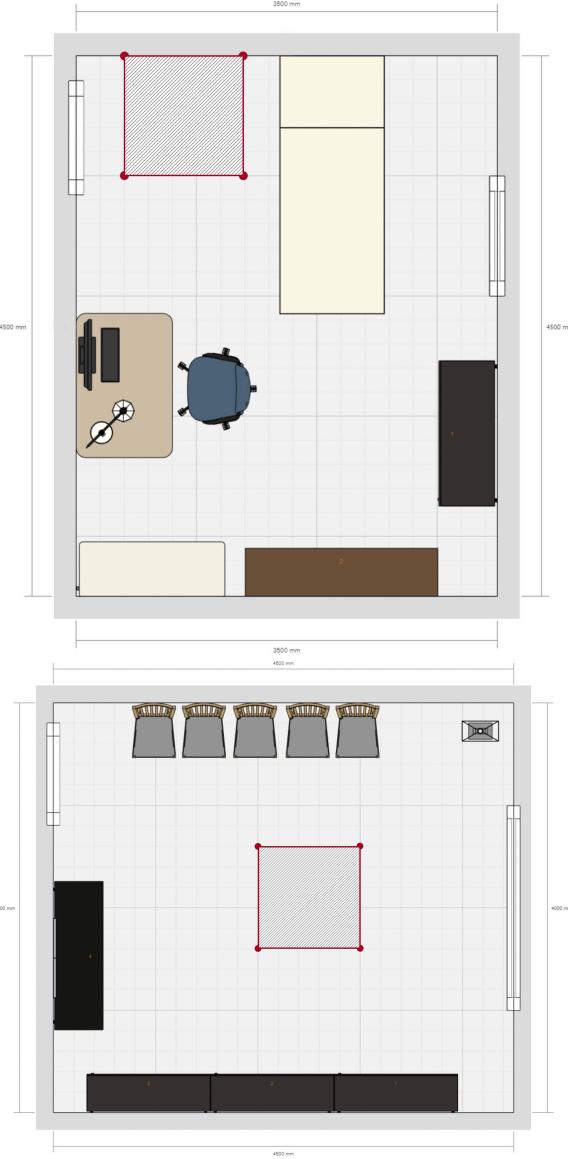


Figure 1. Stanza indoor utilizzate per gli esperimenti a corto raggio, con diverse distanze tra l'area degli esperimenti e i riferimenti attorno.

### 4.1. Corto Raggio

Gli esperimenti a corto raggio sono esperimenti che richiedono al più pochi metri per essere svolti. In particolare abbiamo svolto gli esperimenti indoor in una camera da letto e in un salotto, mentre gli esperimenti outdoor sono stati effettuati su un terrazzo. In ciascun environment abbiamo fatto percorrere alla telecamera un percorso rettilineo di 1 metro, uno quadrato (1 m x 1 m) e uno circolare (cerchio di diametro 1 m), traslandola sul pavimento. Abbiamo cercato di mantenere la velocità costante il più possibile in modo

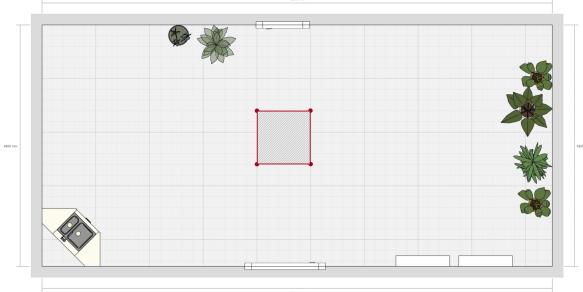


Figure 2. Terrazzo outdoor utilizzato per gli esperimenti a corto e medio raggio.

da poterci concentrare sulla qualità del tracking. Per ogni esperimento, inoltre, abbiamo estrapolato i dati relativi alla posizione, alla velocità e al livello di confidenza del tracking, tralasciando le accelerazioni e le possibili rotazioni. Inoltre, abbiamo svolto un esperimento per testare l'affidabilità del tracking coprendo alternativamente una delle due telecamere e coprendole entrambe. Le cartelle di riferimento per questi esperimenti sono “full\_indoor”, “full\_outdoor”, “shaking”, “indoor\_2d”, “indoor\_2d\_raspi4”, “full\_indoor\_bigger\_room”, “fisheye\_covered” e “white\_wall”.

#### 4.2. Medio Raggio

Gli esperimenti a medio raggio sono quelli contenuti nella cartella “medium\_lines\_back”, e sono stati svolti solo in outdoor sullo stesso terrazzo usato per gli esperimenti a corto raggio. In particolare, abbiamo tenuto conto degli stessi parametri degli esperimenti della categoria precedente, mentre ci siamo differenziati per la distanza percorsa: per effettuare i primi esperimenti riguardo all'affidabilità su percorsi più lunghi, infatti, abbiamo usato due percorsi rettilinei di lunghezza 3m e 6m, tenendo la telecamera ad altezza del petto e camminando con passo regolare in modo da simulare una velocità costante. In ciascun esperimento il tracciato è stato percorso avanti e indietro, tornando all'origine, così da poter testare, oltre all'accuratezza della distanza percorsa, anche la precisione della telecamera nel tracciare il percorso nello spazio.

#### 4.3. Lungo Raggio

In questi environments abbiamo raccolto gli stessi dati degli esperimenti a corto raggio, ma piuttosto che far percorrere alla telecamera un tracciato geometrico regolare, abbiamo testato l'affidabilità dei dati su un percorso arbitrario e per cui fosse facile effettuare un riscontro per mezzo di misurazioni manuali. I percorsi

sono tali da avere punto di origine e punto di arrivo coincidenti e sono stati effettuati in una stanza di grandi dimensioni (come un deposito o un magazzino) con alcuni oggetti al suo interno per l'indoor, e in un giardino con ulivi ed altre piante a distanza ravvicinata per l'outdoor. Gli ambienti per questi esperimenti sono stati scelti in modo da avvicinarsi il più possibile ad un caso d'uso reale, in cui un drone effettua un volo di prova in un giardino o una macchina RC testa algoritmi SLAM in un percorso chiuso ma comunque sufficientemente spazioso. I dati degli esperimenti sono contenuti nella cartella “long”.

#### 4.4. Lunghissimo Raggio

Data la natura dell'esperimento, è ragionevole pensare che quelli che noi consideriamo esperimenti “a lunghissimo raggio” siano compatibili con scenari principalmente outdoor. Uno scenario indoor possibile potrebbe prevedere un hangar di un aeroporto o simili, ma non avendo a disposizione tali spazi, abbiamo effettuato gli esperimenti solo in outdoor. Per questi esperimenti ci siamo avvalsi della “Pista Rossa” a Seano, ossia una pista pensata per attività sportiva lunga circa 500 m. Per questi esperimenti abbiamo testato anche le capacità dell'accelerometro, oltre a tutti i dati già considerati nelle altre categorie. Nello specifico abbiamo percorso la pista sia a piedi che in bicicletta sfruttando un ciclocomputer GPS Garmin Edge 130 per fare un raffronto tra i dati raccolti e poter stimare un errore. In particolare, per gli esperimenti in bicicletta abbiamo effettuato due prove, una a velocità costante per testare il comportamento della telecamera con movimenti più rapidi e una a velocità variabile per testare la precisione nella rilevazione delle accelerazioni. I dati relativi a queste prove sono contenuti nella cartella “very\_long” e “very\_long\_bike”.

#### 4.5. Rotazione

Poiché la telecamera è equipaggiata anche con un giroscopio, abbiamo voluto effettuare una serie di prove a parte per testare la rotazione. I dati sono raccolti nelle cartelle “static\_rotations”, “full\_rotations” e “z\_rotations” e si occupano di testare il corretto rilevamento della variazione degli angoli di pitch, roll e yaw. I test che sono stati effettuati prevedono delle rotazioni intorno agli assi della telecamera mentre questa è tenuta sospesa in aria senza traslazioni, una rotazione con telecamera appoggiata su un oggetto che ruota intorno al suo centro di massa e un esperimento per testare specificatamente l'angolo di yaw intorno all'asse z: in questi esperimenti la videocamera è stata fissata ad una sedia girevole da ufficio con la telecamera rivolta prima dalla parte opposta allo schienale, poi

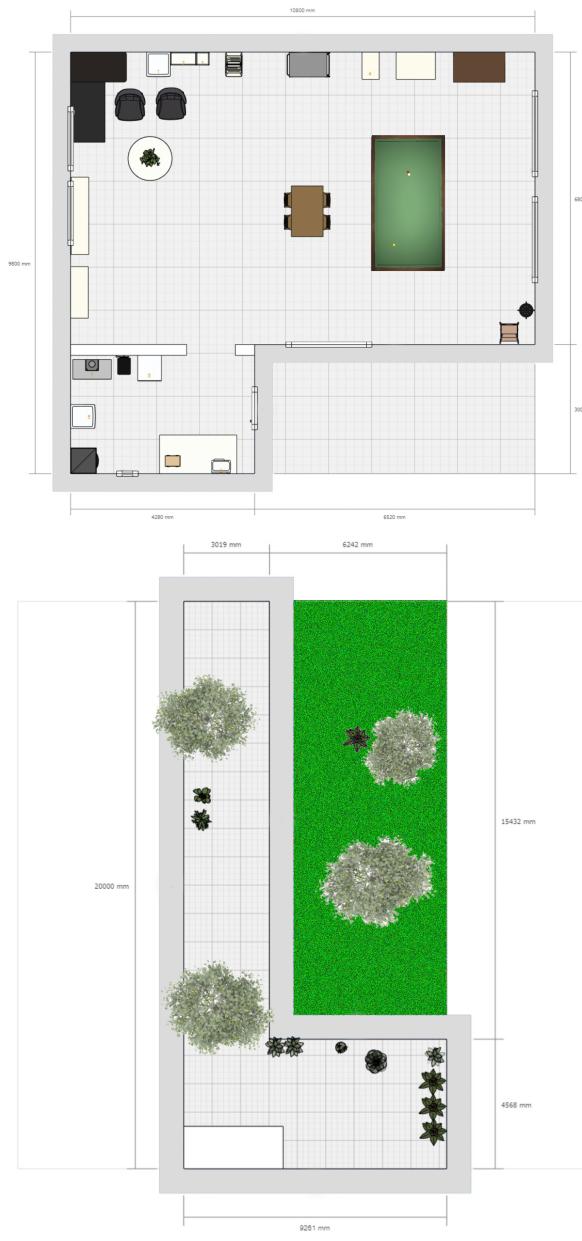


Figure 3. I luoghi scelti (rispettivamente indoor e outdoor, per gli esperimenti a lungo raggio).

rivolta verso di esso, così da simulare l’acquisizione di una rotazione rispettivamente in presenza e in assenza di riferimenti circostanti.

## 5. Risultati

Come nella sezione precedente, per mostrare i risultati ci avvaliamo della stessa divisione in sottosezioni. I risultati verranno presentati sia a livello qualitativo che quantitativo. Un primo risultato generale piuttosto importante è dato dal fatto che la telecamera ha bisogno di un “periodo di stabilizzazione” all’inizio dell’esperimento per ottenere risultati più stabili e precisi. Abbiamo quantificato empiricamente questo intervallo come 5 secondi e, per dare un’idea della sua importanza, riportiamo per ciascuna sezione entrambe le versioni degli esperimenti.



Figure 4. Ciclabile circolare di 500 m, percorsa per effettuare i test a lunghissimo raggio.

tosto importante è dato dal fatto che la telecamera ha bisogno di un “periodo di stabilizzazione” all’inizio dell’esperimento per ottenere risultati più stabili e precisi. Abbiamo quantificato empiricamente questo intervallo come 5 secondi e, per dare un’idea della sua importanza, riportiamo per ciascuna sezione entrambe le versioni degli esperimenti.

### 5.1. Corto Raggio

- **full\_indoor:** Gli esperimenti in questo environment hanno mostrato esattamente ciò che ci aspettavamo: essendo un ambiente con pochi riferimenti e molto vicini, la nostra attesa era di avere errori di stima piuttosto apprezzabili in ogni frangente. Questo comportamento è notabile nelle performance “instabili” della telecamera: alcuni esperimenti funzionano bene, altri funzionano male, ripetere due esperimenti non sempre porta allo stesso risultato e una minima variazione del setup può portare a enormi cambiamenti. Come già detto nella sezione “Esperimenti”, per gli esperimenti indoor a corto raggio abbiamo fatto percorrere alla videocamera tre tracciati, uno rettilineo, uno quadrangolare e uno circolare. Come è possibile vedere dalle figure, gli esperimenti mostrano che in questo environment l’utilizzo del periodo di stabilizzazione non porta grossi vantaggi: sembra peggiorare nel caso del cerchio, che comunque viene piuttosto errato in entrambi i casi, eguagliare nel quadrato e migliorare molto l’accuratezza nelle distanze del percorso rettilineo. A causa dell’errore sul cerchio, abbiamo pensato inoltre di ripetere gli esperimenti facendo ruotare la telecamera lungo la direzione di percorrenza. Possiamo notare che se il quadrato

peggiora visibilmente le performance, nel caso del cerchio queste vengono grandemente migliorate, con un'accuratezza piuttosto buona anche sulle distanze. Per quanto riguarda la confidenza del tracking, possiamo vedere che la telecamera non raggiunge quasi mai una confidenza “alta” ma si attesta sul livello “medio” per la maggior parte del tempo. Gli unici esperimenti in cui il grado massimo viene raggiunto sono quelli con percorso quadrato, e si può vedere (in linea con la nostra attesa) che questo è raggiunto molto prima nell'esperimento con attesa iniziale.

- **full\_outdoor:** In questo environment abbiamo ripetuto gli esperimenti “full\_indoor” ma su un terrazzo. Ciò significa che la telecamera ha a disposizione riferimenti più lontani e, di conseguenza, le sue performance generali dovrebbero migliorare. Ciò è esattamente quello che è successo: gli esperimenti mostrano delle misure generalmente più realistiche, tranne nel caso del percorso rettilineo. Inoltre, in questo environment è ben visibile il miglioramento che comporta l'attesa iniziale di 5 secondi: sebbene le misurazioni siano più o meno comparabili, il tracciato è molto più stabile ed in generale preciso. Ciò che ci ha lasciati un po' interdetti, tuttavia, è il calcolo della confidenza: ci saremmo aspettati che questa fosse maggiore con l'attesa dei 5 secondi, mentre i risultati mostrano che gli esperimenti senza attesa massimizzano la confidenza molto prima degli esperimenti con attesa.
- **shaking:** Questo particolare esperimento è servito per vedere come si comportava la T265 nel caso in cui fosse semplicemente lasciata ferma, appoggiata su un piano. Abbiamo effettuato questi esperimenti nello stesso environment di “full\_indoor”, quindi con riferimenti vicini. I risultati che abbiamo ottenuto sono in linea con quello che ci aspettavamo: la telecamera non riesce a stimare molto bene e registra degli spostamenti, seppur piccoli, comunque non trascurabili, compresi tra i 2 e i 6 cm. Il grado di confidenza è quasi sempre alto nella maggior parte dei casi, e ciò ci ha fatto pensare che in un ambiente con pochi riferimenti, la telecamera tende più facilmente a “fidarsi” dei pochi riferimenti che ha, risultando in confidence score alti anche per rilevazioni sostanzialmente errate.
- **indoor\_2d, indoor\_2d\_raspi4:** Gli esperimenti contenuti in queste due cartelle ci sono serviti in fase iniziale di sperimentazione per vedere quali fossero le differenze tra i modelli 3 e 4

di Raspberry Pi, e sono stati svolti nello stesso ambiente indoor dei risultati a corto raggio. Sebbene non si notino differenze a livello di performance, è stato notato che in generale, a causa della minore capacità computazionale, il Raspberry Pi 3 era più soggetto a freeze e crash qualora si tentasse di utilizzare il software realsense-viewer di Intel. Ciò ci ha spronato ad usare solamente il Raspberry Pi 4 per i successivi esperimenti.

- **full\_indoor\_bigger\_room:** In questo environment abbiamo ripetuto gli esperimenti in una stanza più grande, in modo tale da verificare che riferimenti più diversificati e più lontani risultassero sostanzialmente in migliori performance. Dagli esperimenti ciò risulta vero ed è ben visibile nelle geometrie disegnate nei grafici. Notabile come anche i confidence scores risultino in generale migliori e comparabili sia con che senza i 5 secondi di attesa iniziale e che il girare la telecamera lungo la direzione del moto fornisce un tracking comunque preciso ma con confidenza generalmente più bassa. Ciò è interpretabile secondo noi con il fatto che se la telecamera non viene ruotata, tende ad essere più fiduciosa ma anche più prona ad errori, mentre il cambiamento di riferimenti fornisce una confidenza minore ma più realistica.
- **fisheye\_covered:** Con queste prove abbiamo voluto testare il grado di affidabilità in condizione di visione con una singola camera. Abbiamo effettuato il test a corto raggio su percorso rettilineo tappando alternativamente una delle due telecamere, e successivamente le abbiamo tappate entrambe. Non ci sono molte differenze nel caso di una telecamera tappata, mentre la libreria ci restituisce un errore nel caso di due telecamere occluse. Anche per quanto riguarda i confidence scores, vediamo che in tutti e tra i casi la telecamera non riesce a superare la confidenza “media”.
- **white\_wall:** Per verificare l'importanza delle telecamere all'interno del software della RealSense, abbiamo provato a confonderle un po' dandole un muro bianco come unico riferimento. In particolare nel primo esperimento abbiamo percorso un tratto di stanza con le camere fisse sul muro e ruotato la camera in corrispondenza dell'angolo, mentre nel secondo esperimento si è percorso un tratto di corridoio di circa 2.5 m due volte: una con la camera fissa sulla parete è bianca, mentre la seconda volta con la camera rivolta verso la parete opposta, ricoperta da scaffali. I risultati dimostrano come l'angolo della stanza, nel primo

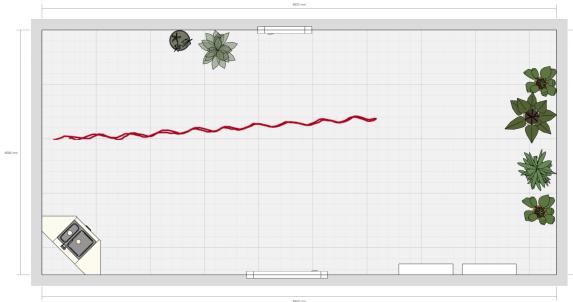


Figure 5. Il plotting della traiettoria nell'esperimento a medio raggio.

caso, non sia stato proprio riconosciuto, mentre nel secondo esperimento, i risultati con il solo muro bianco sono stati peggiori.

## 5.2. Medio Raggio

- **medium\_lines\_back:** Gli esperimenti in questa cartella si sono svolti su un terrazzo, all'aperto, come in “full\_outdoor”, ma con una sostanziale differenza: invece di far strisciare la telecamera lungo il pavimento, in questo caso è stata percorsa una traiettoria rettilinea a piedi con la telecamera ad altezza del petto. Quello che ci aspettavamo era un set di riferimenti a grandissima distanza e quindi un tracking molto preciso. I risultati mostrano che la teoria viene confermata: i percorsi da 3m e 6m vengono tracciati molto bene e, nelle traiettorie, è addirittura visibile lo scostamento perpendicolare dovuto allo spostamento di peso mentre si effettua un passo. Anche per quanto riguarda la confidenza vediamo che un confidence score “alto” viene raggiunto quasi, a dimostrazione che riferimenti molto lontani permettono alla telecamera di effettuare un tracking migliore.

## 5.3. Lungo Raggio

- **long:** Da questi esperimenti ci aspettavamo che fosse evidente nel tracking l'utilità dei 5 secondi di assestamento della videocamera, ed infatti così è stato: essendo i percorsi compresi lunghi complessivamente tra i 50 e i 100 m, pensavamo che l'errore di assestamento (che empiricamente si assiste tra il 2% e il 5% del percorso) fosse nell'ordine dei 3-5 m, invece che di pochi centimetri. Ed infatti così è stato: tenendo conto che i percorsi hanno inizio e fine coincidenti, è ben visibile che, nonostante gli score sulla confidenza risultino migliori nel caso senza attesa, il tracking risulta decisamente migliore se si attendono i 5 secondi di

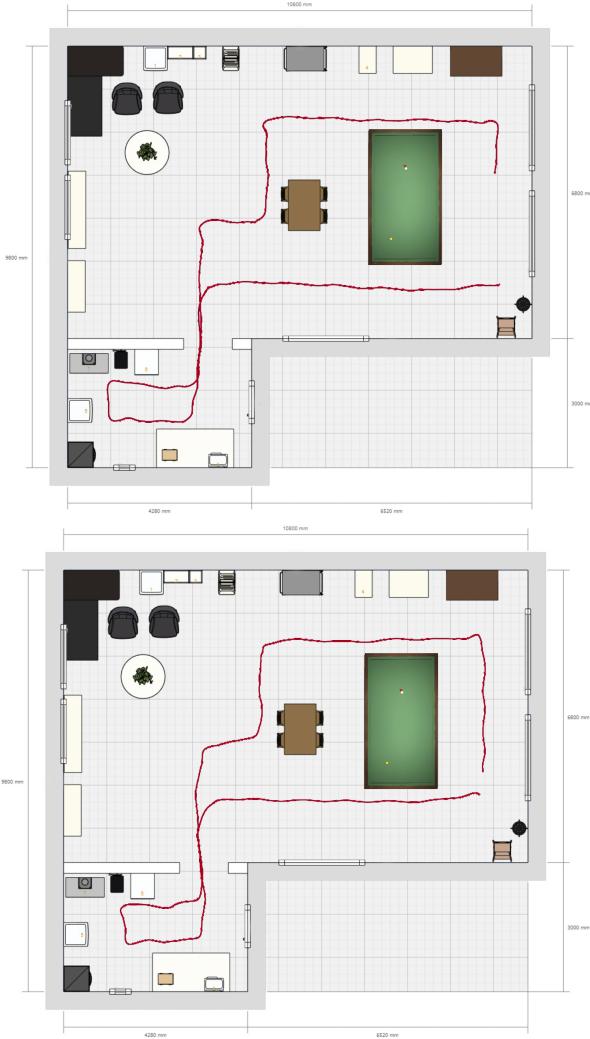


Figure 6. Il plotting della traiettoria rilevata nell'ambiente: si noti, nella seconda immagine, come i 5 secondi preliminari abbiano migliorato la precisione nel primo tratto.

inizializzazione. I risultati sulla confidenza, uniti a quelli a corto raggio, hanno cominciato a farci pensare che in alcune situazioni la telecamera diventi troppo “sicura di sé” quando non dovrebbe.

## 5.4. Lunghissimo Raggio

- **very\_long:** L'esperimento condotto sulla pista all'esterno ci ha permesso di focalizzarci su situazioni comuni in cui la telecamera potrebbe essere utilizzata, come ad esempio l'utilizzo su un drone, una macchina o una bicicletta. La nostra attesa era che la telecamera riuscisse a tracciare molto bene il percorso e che potesse in qualche modo rendere al meglio le sue potenzialità. Tali aspet-

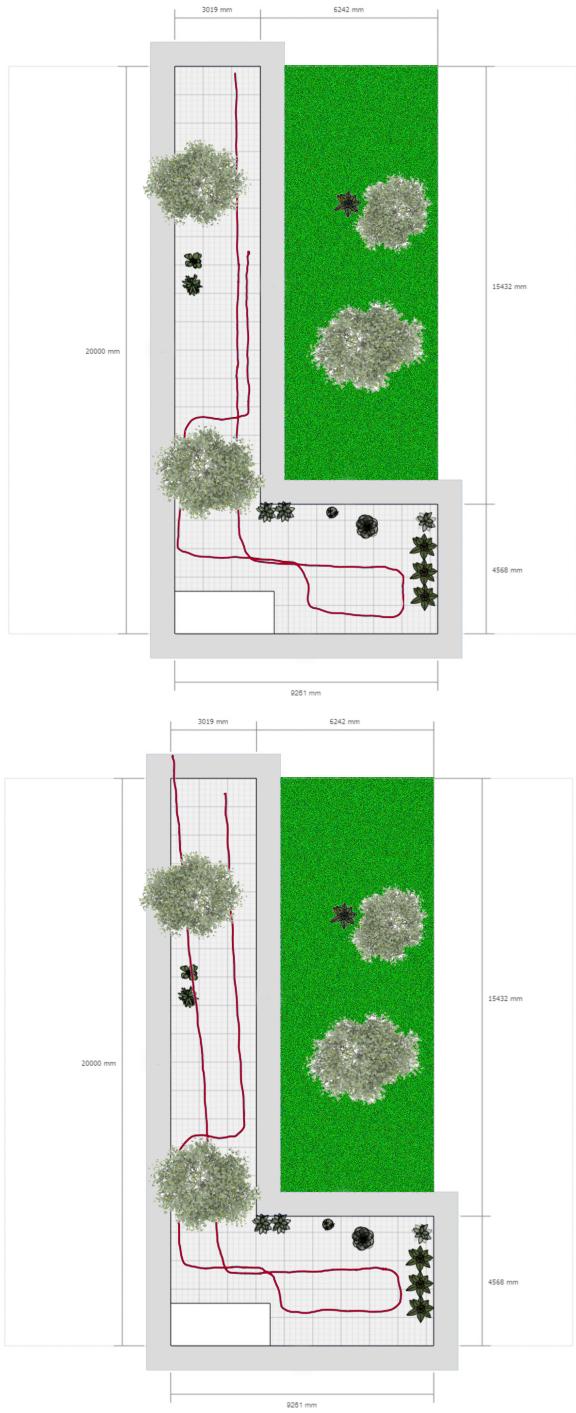


Figure 7. Il plotting della traiettoria in ambiente esterno. Come si evince dalla seconda immagine, i 5 secondi hanno aiutato la rilevazione, anche se nell'ultimo tratto si è perso un po' di parallelismo tra le due traiettorie, probabilmente a causa di imprecisioni dovute agli alberi.

tative sono state in parte soddisfatte ed in parte

deluse: vediamo ad esempio come il tracking segua quasi esattamente il percorso effettivamente fatto, ma avendo a disposizione le misure della pista possiamo vedere come alcune di queste misure non combacino. L'errore si attesta comunque sotto il 10%. Un'altra delusione è derivante dal fatto che nonostante fosse un circuito, inizio e fine del tracking non corrispondano, ed anzi che ci sia un errore notabile di circa 15m (su circa 100m di raggio della pista). Il nostro pensiero è che questo errore sia dovuto al fatto che la telecamera, internamente, integri i dati sulla velocità per ottenere la posizione e dunque ottenga un effetto di "drifting" che sulle lunghe distanze risulta visibile. D'altra parte è possibile vedere come, in un ambiente del genere, la confidenza della telecamera risulti praticamente sempre alta, e ci sentiamo di dire che in questo frangente non sembra essere dovuta alla "tropica confidenza".

- **very\_long\_bike:** Visto il ragionamento del punto precedente, ci siamo chiesti che cosa potesse succedere alle rilevazioni se si aumenta la velocità e soprattutto se questa viene variata. Inoltre, volendo testare anche l'accelerometro, ci siamo sentiti in dovere di fare un esperimento analogo al precedente, ma invece di mantenere velocità costante lungo tutta la pista, di usare una bicicletta e percorrerla a velocità variabile (e in media più elevata). La nostra idea era che a causa della velocità più alta la telecamera avrebbe dovuto fare più fatica a stimare la propria velocità e quindi, se l'ipotesi dell'integrazione fosse stata vera, portare significativi errori nel tracking. Questa ipotesi si è rivelata corretta in quanto, guardando il tracking, si nota subito come questo si discosti molto da quello originale, pur mantenendo una rassomiglianza con il percorso effettivo. Inoltre, è stato effettuato un test su più giri in bicicletta a velocità costante, per effettuare un confronto con la versione dell'esperimento precedente (condotta a piedi). Anche in questo caso, data la maggiore velocità, si nota come l'errore di drifting viene accentuato, ad ulteriore conferma della nostra ipotesi, generando dei tracciati che solo vagamente ricordano il tracciato realmente percorso. Per quanto riguarda la confidence, abbiamo interpretato la confidenza altissima come corretta in quanto simile a quella dell'esperimento precedente, e abbiamo attribuito l'errore alla variabilità della velocità che rendono dunque i calcoli integrali decisamente più complessi e quindi più imprecisi. Su questo esperimento, inoltre, abbiamo testato la sensibilità sulle variazioni di velocità ed accel-

erazioni e ne abbiamo tracciato i grafici spazio-tempo e velocità tempo, come visibile. Inoltre, abbiamo affiancato un GPS Garmin alla camera in modo da poter avere un termine di paragone affidabile sul tracking a lungo raggio. Abbiamo riscontrato che le velocità risultano generalmente accurate e rispecchiano l'andamento reale, con qualche incertezza lungo le curve, ma spiegabili col fatto che in quei tratti agiscono più forze e che il risultato finale della camera tiene conto anche di queste componenti. Per quanto riguarda la traccia, invece, quella del Garmin risulta essere molto più credibile, soprattutto per quanto riguarda l'andamento su più giri, contesto che il RealSense soffre di più. Un dispositivo GPS, anche se nei brevi tratti risulta essere approssimativo, ha la caratteristica di aggiornamento delle coordinate, il che lo rende molto affidabile nelle tracce lunghe. D'altra parte, un dispositivo di tracking risulta essere più preciso, ma una volta compiuto ogni singolo errore, la stima è ormai compromessa e quindi il risultato finale si discosta molto da quello desiderato in misurazioni lunghe. Sarebbe interessante cercare di sviluppare una tecnologia ibrida capace di mitigare i difetti di ognuna delle tecnologie e sommarne i vantaggi.

## 5.5. Rotazione

- **static\_rotations:** Gli esperimenti di questa cartella ci sono serviti principalmente per comprendere il grado di precisione delle rotazioni lungo gli assi. Per ottenere i dati abbiamo appoggiato la camera su una superficie rigida e parallela al terreno e abbiamo raccolto dati per una ventina di secondi. Come è possibile vedere dai grafici, le rotazioni sono minime e sono probabilmente dovute all'environment poco adatto (si tratta infatti dello stesso ambiente degli esperimenti indoor a corto raggio). I confidence scores sono realistici e riflettono il comportamento degli altri esperimenti nelle stesse condizioni: pochi riferimenti rendono difficile il raggiungimento di un grado di confidenza alto.
- **full\_rotations:** I dati raccolti per questo esperimento ci forniscono una panoramica sulla precisione del tracciamento delle rotazioni. Queste sono state fatte ruotando la telecamera intorno ai suoi tre assi in maniera più precisa possibile. Dai grafici è possibile vedere che le rotazioni lungo gli assi x e y sono molto precise e rispecchiano l'andamento atteso, per cui possiamo dire che lungo questi due assi il tracking è molto preciso. Tutt'altra storia



Figure 8. Il plotting della traiettoria rilevata sulla ciclabile: la prima figura rappresenta quella con RealSense T265 a piedi, mentre la seconda è stata rilevata in bicicletta tramite Garmin.

si verifica nel caso della rotazione lungo l'asse z: in questo caso il tracciamento è totalmente scorretto e anzi non si riesce nemmeno a intravedere un pattern nell'errore. L'ipotesi che abbiamo formulato è che per le rotazioni sull'asse z venissero usati anche dei dati presi dalle telecamere e che, dato l'ambiente ostico, questo generasse un errore. Ripetendo gli esperimenti abbiamo inoltre visto che anche in questo modo si ottengono buoni risultati, a patto che si usi una velocità di rotazione sufficientemente elevata.

- **z\_rotations:** Data la nostra ipotesi sulla rotazione lungo l'asse z, abbiamo pensato di dover effettuare un esperimento a parte per testare la fun-

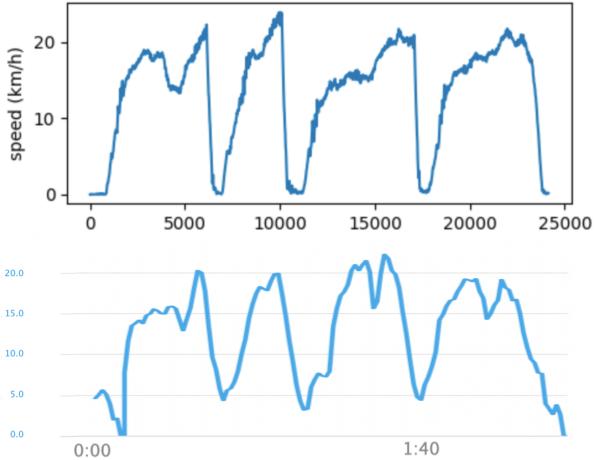


Figure 9. Il plotting delle velocità rilevate durante un giro di pista, in particolare il grafico soprafigurante raffigura i dati della RealSense, mentre in basso i corrispondenti dati del Garmin. I primi hanno un andamento più dettagliato, probabilmente più sensibile rispetto al rilevamento della velocità istantanea tramite GPS, che risulta essere più smooth.

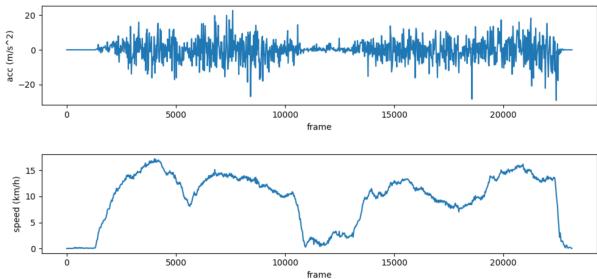


Figure 10. Il plotting dell'accelerazione rilevata durante un giro di pista ciclabile dalla RealSense e le velocità relative.

zionalità della telecamera su questo frangente. Inoltre, siccome le rotazioni lungo l'asse z sono le più comuni a cui la camera è sottoposta nei suoi scenari d'uso tipici, abbiamo deciso di approfondire un po' di più con due prove separate. Dai grafici è possibile vedere che praticamente tutti gli andamenti riportano dei confidence score “medi” come ci aspettavamo, e vediamo che sia nel caso di telecamera rivolta verso lo schienale che nel caso opposto il tracking viene effettuato correttamente sia su uno che su più giri e utilizzando diverse velocità di rotazione. Come già accennato sopra, rotazioni più lente portano a risultati meno accurati, ma comunque accettabili. Il caso con telecamera rivolta verso lo schienale ci ha sorpresi, ma immaginiamo che tale risultato sia dovuto al grande angolo di visione che le telecamere fisheye riescono a catturare e la limitata estensione dello schienale della

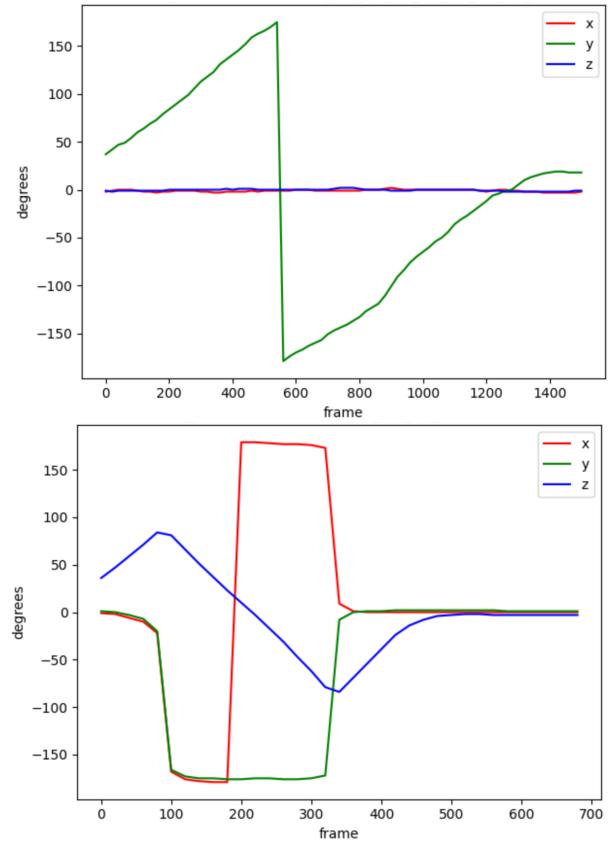
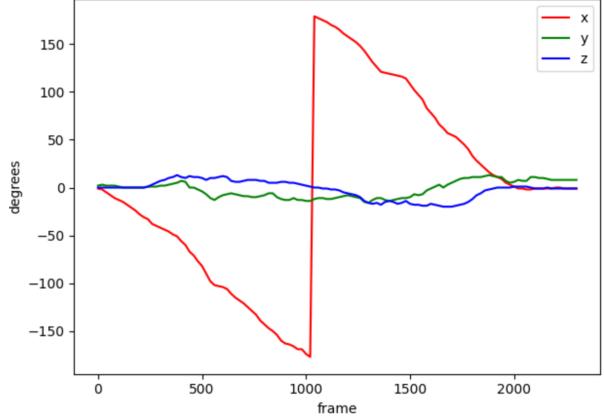


Figure 11. Il plotting dell'angolo durante il tempo di rotazione completa di roll, pitch e yaw. La rotazione lungo l'asse z è meno precisa quando si ruota la camera a basse velocità.

sedie, che lasciava inevitabilmente intravedere una piccola parte di sfondo.

## 6. Conclusioni e Sviluppi Futuri

Dati gli esperimenti condotti e visti i loro risultati, possiamo trarre alcune interessanti conclusioni:

- **Le caratteristiche dell’ambiente sono importanti.** I risultati ottenuti dalla camera sono fortemente dipendenti dall’environment utilizzato e spaziano da risultati ottimi (con errore molto piccolo) a risultati appena accettabili o addirittura inaccettabili in condizioni particolarmente limitanti;
- **Aspettare qualche secondo per l’assestamento della camera.** La telecamera funziona in generale meglio quando si attendono 5-10 secondi prima del suo utilizzo. Sebbene questa attesa non sempre porta ad un miglioramento, non peggiora mai le prestazioni, ed è di conseguenza da considerare una buona soluzione da provare in un environment particolarmente ostico;
- **Attenzione al drifting su percorsi lunghi.** Il drifting della telecamera nel tracking è inevitabile e si accumula con l’aumentare del tempo di tracciamento, in quanto probabilmente dovuto ad errori di integrazione dei rilevamenti dei sensori di velocità e pertanto completamente indipendenti dall’ambiente di utilizzo. Nel caso di utilizzo in scenari che richiedono precisione, potrebbe essere utile provare a stimare questo errore in maniera più accurata e magari costruire un modello correttivo. Una possibile soluzione nell’immediato potrebbe consistere nell’affiancare la camera ad un GPS, come visto negli esperimenti, e utilizzare una combinazione dei dati per rimediare all’errore;
- **Controllare i confidence scores.** In alcuni ambienti di test particolarmente difficili abbiamo notato che la telecamera tende ad aumentare i suoi confidence scores in modo apparentemente immotivato, risultando in performance molto diverse dalla realtà supportate però da un’alta confidenza. Un possibile lavoro futuro potrebbe approfondire questo aspetto e cercare studiarne una soluzione;
- **Ruotare lo yaw a velocità adeguate.** Per quanto concerne le rotazioni, una rotazione lungo l’asse x o y non rappresenta un problema in nessun ambiente, mentre rotazioni lungo l’asse z richiedono una velocità angolare minima per funzionare correttamente. Per un utilizzo con rotazioni a bassa velocità potrebbero essere necessari ulteriori studi per comprendere meglio questo fenomeno;
- **Affidarsi alle rilevazioni di velocità e accelerazione.** In generale la telecamera rileva correttamente le accelerazioni e le velocità, individuando anche le minime variazioni di velocità. Confrontando anche con un sistema GPS, RealSense

sembra essere localmente più precisa. Come già accennato sopra, unire un dispositivo di tracking che non è affatto da drifting (ad esempio il GPS) potrebbe costituire un’ottima *baseline* per un approfondimento futuro.

In generale, la Intel RealSense T265 svolge un buon lavoro in svariati contesti ma, date le prove effettuate, raggiunge l’eccellenza solo in ambienti aperti o comunque spaziosi e ricchi di riferimenti.

Gli esperimenti sono stati possibili grazie agli strumenti forniti dall’Università di Firenze sotto la supervisione del Prof. Michele Basso e del Dott. Luca Bigazzi.

## References

- [1] Datasheet ufficiale di Intel RealSense T265  
[https://www.intelrealsense.com/wp-content/uploads/2019/09/Intel\\_RealSense\\_Tracking\\_Camera\\_Datasheet\\_Rev004\\_release.pdf](https://www.intelrealsense.com/wp-content/uploads/2019/09/Intel_RealSense_Tracking_Camera_Datasheet_Rev004_release.pdf)
- [2] Pagina ufficiale di Raspberry,  
<https://www.raspberrypi.org/>
- [3] Intel RealSense SDK  
<https://www.intelrealsense.com/developers/>
- [4] Guida ufficiale per la compilazione e l’installazione di Intel RealSense SDK su Raspbian,  
[https://github.com/IntelRealSense/librealsense/blob/master/doc/installation\\_raspbian.md](https://github.com/IntelRealSense/librealsense/blob/master/doc/installation_raspbian.md)
- [5] Documentazione della libreria PyRealSense2,  
[https://intelrealsense.github.io/librealsense/python\\_docs/\\_generated/pyrealsense2.html](https://intelrealsense.github.io/librealsense/python_docs/_generated/pyrealsense2.html)
- [6] Drivers per utilizzare schermo touchscreen LCD esterno,  
<https://github.com/goodtft/LCD-show>
- [7] Repository GitHub del progetto  
[https://github.com/FrancescoGradi/RealSense\\_T265\\_evaluator](https://github.com/FrancescoGradi/RealSense_T265_evaluator)

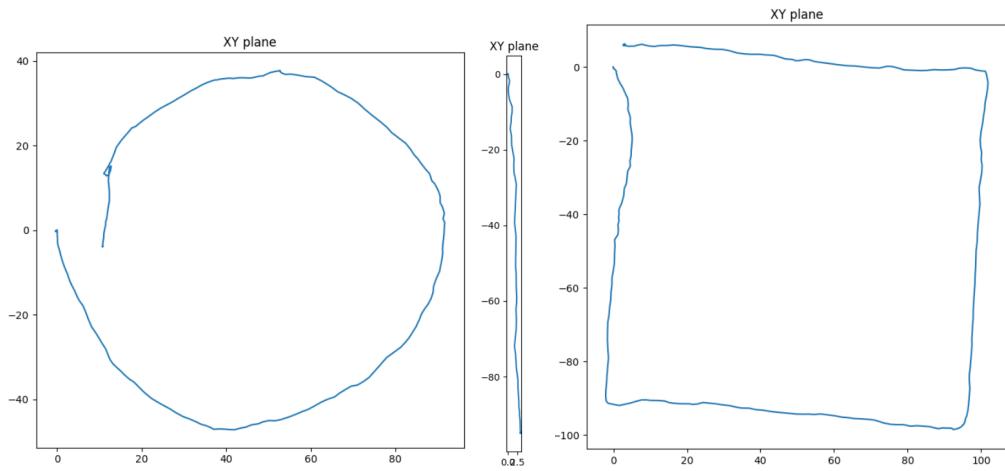


Figure 12. Grafici indoor nel primo scenario con riferimenti molto vicini.

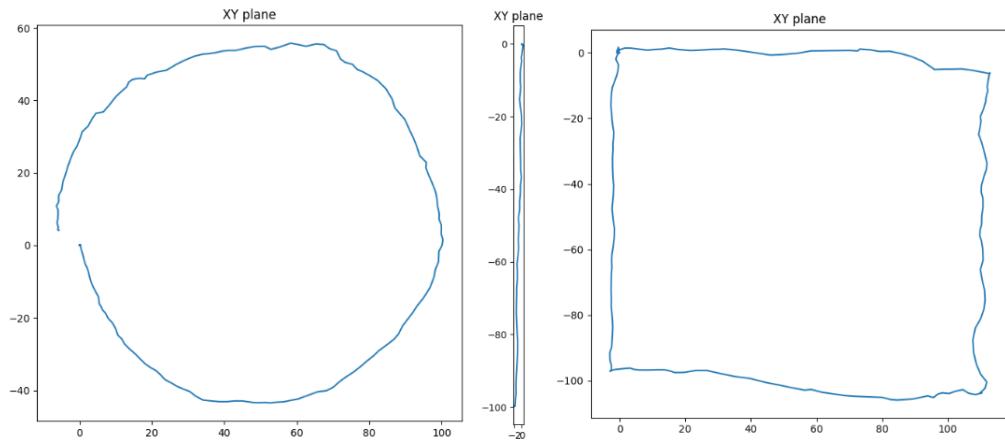


Figure 13. Grafici outdoor con riferimenti più lontani. Si notano risultati migliori

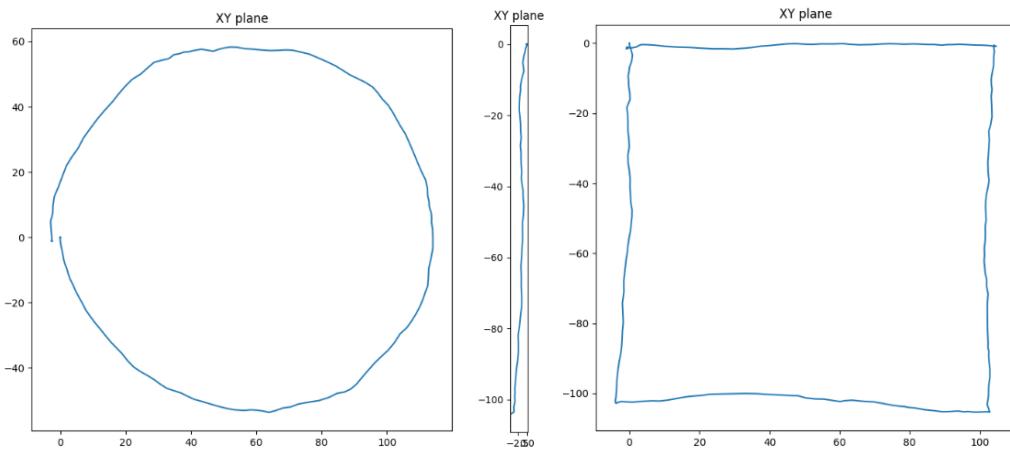


Figure 14. Grafici indoor nella stanza più grande con riferimenti a distanza media. Si notano risultati ottimi.

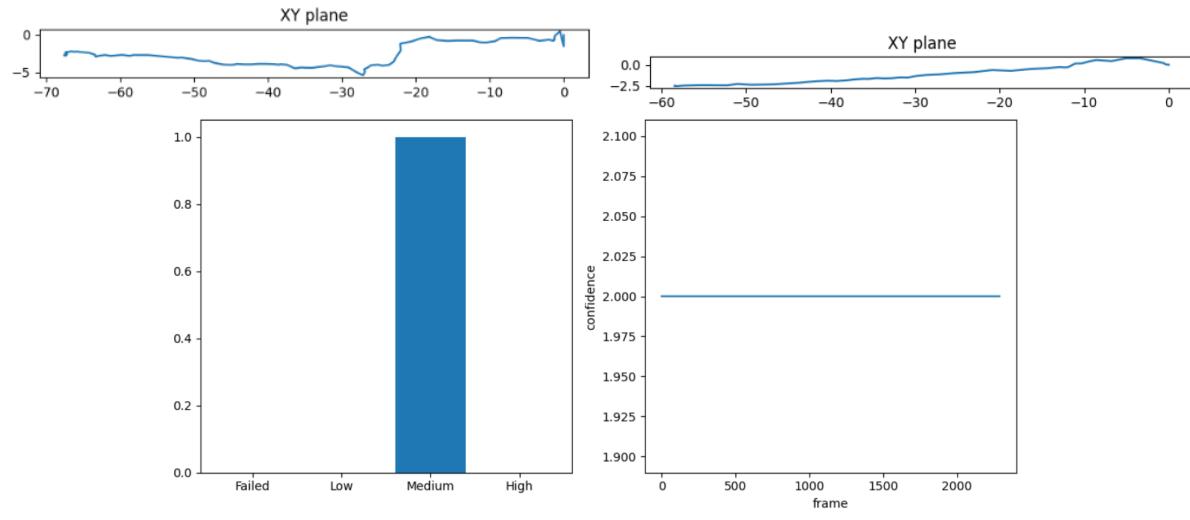


Figure 15. Grafici indoor con una fisheye occlusa (prima linea) e senza. Si notano peggioramenti nel primo caso, con confidenza sempre media.

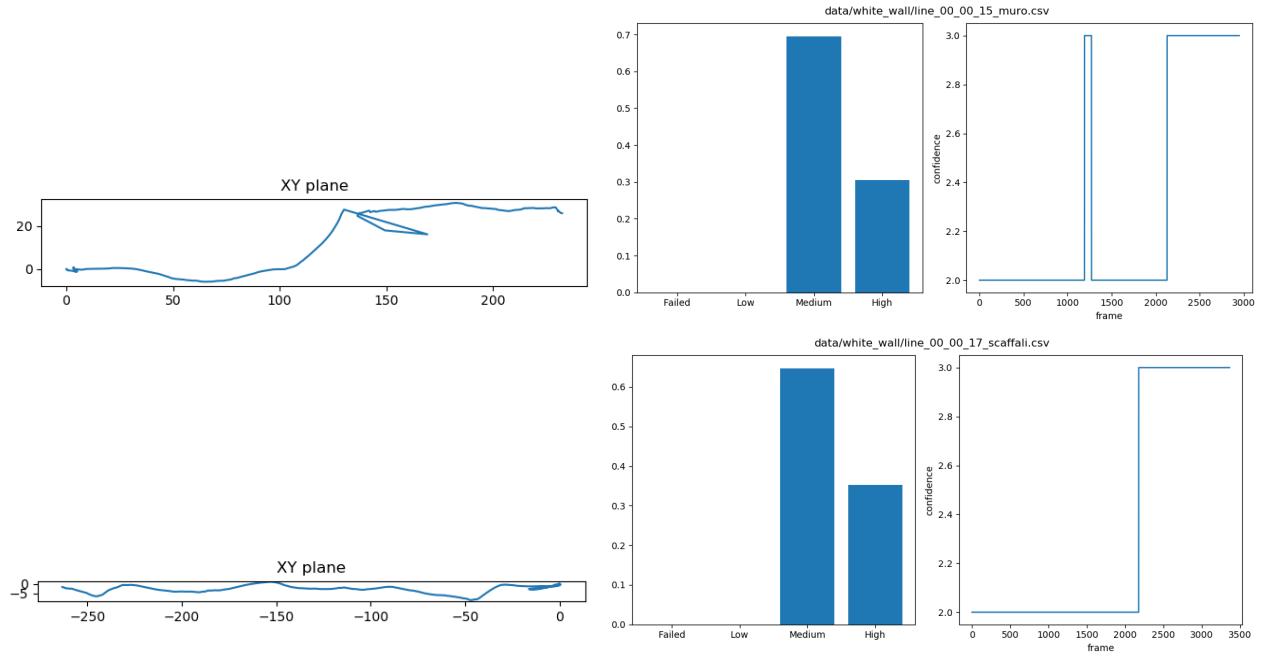


Figure 16. Grafici indoor: il primo rappresenta il tracking con camera che trasla osservando un muro bianco, mentre il secondo trasla osservando degli scaffali. Si noti come nel primo caso la rilevazione sia stata ben peggiore.

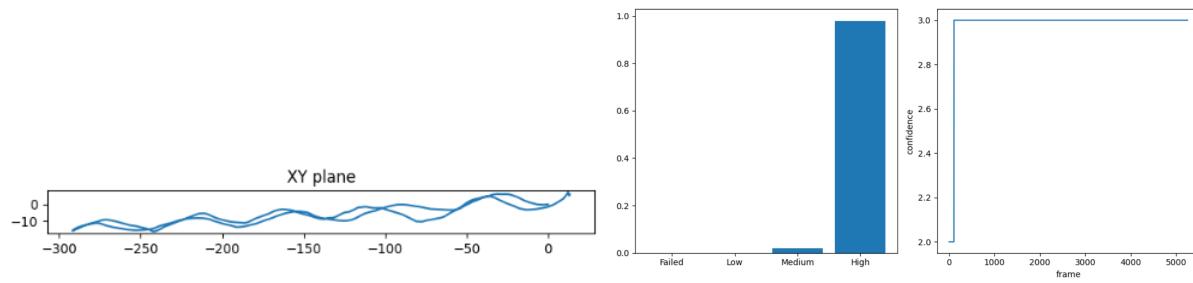


Figure 17. Grafici outdoor percorrendo una linea di 3 metri avanti e indietro: è possibile notare la precisione nella rilevazione delle oscillazioni dei passi.