

Rilevamento e Localizzazione di Persone tramite Kinect, GPS, YOLO e bussola, con Data Fusion tramite rete neurale

1. Introduzione

Il progetto si è concentrato sul migliorare e integrare il progetto precedente, aggiungendo ai sensori già presenti (quello RGB, di profondità del Kinect e il segnale GPS) una bussola per poter calcolare la posizione globale dei rilevamenti. Tutti i dati vengono poi uniti ed elaborati attraverso un neural network con implementazione di LSTM.

L'obiettivo principale è stato, diversamente dai progetti precedenti, quello di sviluppare un'implementazione di principio di un sistema di rilevamento dei volti e di misurazione della distanza in tempo reale che applica data fusion per determinare la possibile posizione globale dei volti individuati. Questa scelta è stata dettata dal fatto che, per addestrare il modello da noi progettato, servirebbe un dataset di buone dimensioni, mentre il nostro era decisamente ridotto, e portava comunque un ciclo di addestramento (10 epoche) a durare circa 5 ore.

2. Librerie, algoritmi e framework utilizzati

- **YOLO11**: L'ultimo modello di deep learning di Ultralytics per il rilevamento rapido di oggetti, addestrato su un dataset personalizzato
- **MEGAFACE**: Dataset contenente le immagini di volti/persona per il training
- **Free Viewpoint RGB-D Video Dataset**: Dataset contenente dati rgb e di profondità per il training
- **PyTorch**: Framework per il training di YOLO e del neural network custom
- **Freenect**: Libreria per interfacciarsi con il Kinect
- **OpenCV**: Utilizzato per il pre-processing delle immagini e per l'output
- **NumPy**: Utilizzato per la gestione di matrici e operazioni numeriche
- **Folium**: Libreria per la creazione di mappe interattive

3. Struttura del sistema

Il sistema è composto da diverse fasi e supporta sia l'elaborazione real-time che la post-elaborazione.

L'elaborazione **real-time** è composta dalle seguenti fasi:

- **Calibrazione**: Il processo di calibrazione delle telecamere del sensore Kinect è basato sull'acquisizione di immagini di una scacchiera posta in diverse posizioni davanti al dispositivo. La scacchiera viene utilizzata dal Kinect come riferimento fisso in quanto i punti d'angolo sono disposti in un pattern regolare. Durante la calibrazione, il sistema rileva gli angoli della scacchiera nelle immagini RGB e di depth, associando ogni punto rilevato alle coordinate corrispondenti nello

spazio. Una volta raccolte le immagini, un algoritmo calcola le matrici intrinseche delle telecamere e i loro coefficienti di distorsione, necessari per correggere l'immagine e migliorare la precisione della percezione spaziale. I parametri ottenuti vengono poi salvati e possono essere usati successivamente per migliorare l'allineamento tra immagini e dati di profondità.

- **Acquisizione delle Immagini RGB, della Mappa di Profondità, del segnale GPS e dei dati della bussola:** In questa fase i sensori del Kinect catturano i dati RGB e di profondità, mentre il GPS i dati di posizione globali e la bussola quelli di direzione
- **Rilevamento degli Oggetti con YOLO:** Una volta acquisita l'immagine, questa viene preparata per essere processata dal modello YOLO. Nel codice viene caricato il modello "last.pt" e le classi degli oggetti "obj.names".
- **Passaggio dei dati alla rete e elaborazione:** I dati vengono trasformati in tensori, messi in una lista frame per frame (massimo 10 frame) e passati al modello, che ne estrae le feature frame per frame, poi le concatena e restituisce la sua stima
- **Visualizzazione e Feedback:** Gli oggetti rilevati vengono visualizzati sull'immagine RGB con un bounding box, il loro ID e la distanza stimata dalla rete. In aggiunta ad ogni frame viene generato e riscritto un file html contenente una mappa centrata nel segnale gps, e tanti punti di interesse quanti gli oggetti rilevati, nella loro possibile posizione nello spazio.

La **post-elaborazione** invece verte sulle seguenti fasi:

- **Acquisizione delle immagini RGB, della Mappa di Profondità, del segnale GPS e i dati della bussola:** In questa fase il programma carica il video RGB, il video di profondità, il segnale GPS (contenuto in un file) e i dati della bussola (anch'essi contenuti in un file).

Il **Rilevamento degli oggetti con YOLO**, il **Passaggio dei dati alla rete e elaborazione** e la **Visualizzazione e Feedback** sono i medesimi dell'elaborazione real-time, e in più vengono salvati i risultati in un file CSV.

4. Problematiche riscontrate

Nonostante questo progetto mostri dei dati abbastanza promettenti, è ben lontano dall'essere perfetto. L'addestramento non sufficiente a causa del dataset si vede al momento della predizione. Dati che avevamo cercato di far intendere dalla rete come secondari in questo progetto statico (la camera è fissa e non si sposta), hanno ancora un impatto notevole sulle predizioni della rete. Il cambiamento di orientamento o di posizione (sullo stesso file di valutazione) porta a risultati diversi, ma tutti in un range ammissibile. Le immagini "distance_plot_0-1-2" mostrano questo variare all'aumento dei gradi di orientamento [10.2°, 189°, 358°]

5. Conclusioni

Questo progetto integra il progetto precedente aggiungendogli dati da una bussola e un meccanismo di data fusion tramite rete neurale. Nonostante i dati di valutazione mostrino una necessità di un dataset più esteso e decisamente più training, porta risultati promettenti sulle possibili capacità che potrebbe avere senza le mancanze che ha in questo momento.