Report del 3° progetto di 3D Perception, Learning Based Data Fusion

Autori: Francesco De Patre, Davide Faroldi Lo Presti

Rilevamento e Localizzazione di volti umani tramite tecniche di Data Fusion basate sull'Apprendimento

Introduzione

[Descrizione dell'Obiettivo del progetto e breve spiegazione del suo funzionamento]

Questo progetto si è focalizzato sul miglioramento e sull'integrazione del lavoro precedente, ampliando il sistema sensoriale del Kinect, già composto da sensori RGB, di profondità e dal GPS, con una bussola per calcolare la posizione globale dei rilevamenti in modo più accurato.
I dati raccolti da questi sensori vengono combinati e elaborati tramite una rete neurale che incorpora un modulo LSTM (Long Short-Term Memory), consentendo un'elaborazione temporale dei dati e un miglioramento nella stima delle posizioni.
A differenza dei progetti precedenti, l'obiettivo principale è stato lo sviluppo di una proof-of-concept per un sistema di rilevamento dei volti e misurazione della distanza in tempo reale, con fusione dei dati per determinare la posizione globale dei volti rilevati.
Questa impostazione riflette una sfida notevole: mentre il sistema progettato è funzionale, un addestramento ottimale del modello richiederebbe un dataset più ampio per migliorare le prestazioni e la precisione delle previsioni.
Attualmente, infatti, il dataset limitato a nostra disposizione, pur permettendo un addestramento su 10 epoche, richiede circa 5 ore per completare ogni ciclo di training.
Il codice del nostro progetto è suddiviso in tre versioni principali, ciascuna con uno scopo specifico.
Le prime due versioni sono dedicate all'elaborazione in tempo reale: una utilizza i dati provenienti dai sensori e l'altra opera senza di essi.
La terza versione, invece, è stata sviluppata per il post-processing, consentendo un'analisi approfondita dei dati raccolti.
Ogni versione si avvale di moduli dedicati per la gestione dei sensori e del modello multimodale, ottimizzando l'integrazione e la gestione dei diversi componenti del sistema.

Architettura

[Architettura del progetto e componenti principali]

Le componenti principali che sono state utilizzate nel progetto sono diverse:

- **YOLOV11:** Abbiamo utilizzato l'ultima versione del modello YOLO di Ultralytics per il rilevamento rapido e preciso di oggetti, addestrato su un dataset personalizzato per la rilevazione di volti e persone. YOLO è stato scelto per la sua elevata velocità e accuratezza, particolarmente adatto a scenari di rilevamento in tempo reale.
- **Free Viewpoint RGB-D Video Dataset**: Dataset di dati rgb e depth sincronizzati per il riconoscimento e localizzazione di volti, purtroppo le dimensioni del dataset sono sufficienti solo per una dimostrazione.

- **Pytorch**: Pytorchè stato scelto come framework principale per il training del modello YOLO e della nostra rete neurale custom basata su LSTM per il data fusion. La sua flessibilità e supporto per l'elaborazione su GPU hanno facilitato l'implementazione e l'addestramento dei modelli di deep learning necessari per il progetto.
- **Freenect:** Per l'interazione con i sensori.
- **Folium:** Libreria per la creazione di mappe interattive.
- **OpenCV:** Libreria per il processing delle immagini.
- **NumPy:** Libreria utilizzata principalmente per la gestione di operazioni con matrici.

Il codice è organizzato come un sistema di data fusion che integra rilevamenti di volti e distanze in tempo reale utilizzando sensori Kinect, GPS e bussola.
L'architettura è strutturata in vari moduli che collaborano per raccogliere, processare e visualizzare i dati.
Dopo aver importato librerie fondamentali, come YOLO e pyTorchper l'inferenza con il modello multimodale, il sistema apre i flussi video RGB e di profondità registrati, insieme ai dati di latitudine e longitudine forniti dal GPS.
La mappa di profondità viene convertita in metri per calcolare la distanza effettiva dai punti rilevati e le immagini sono pre-processate per l'inferenza YOLO, che localizza i volti rilevati. Per ciascun volto, viene calcolata la distanza reale e l'angolo di orientamento rispetto al centro della telecamera.
Successivamente, un modello LSTM multimodale riceve come input l'immagine ritagliata del volto e i dati numerici (distanza reale, latitudine, longitudine e angolo) per predire la distanza rispetto alla telecamera.
Le distanze reali e predette vengono memorizzate in un file CSV per analisi future, e una mappa interattiva mostra la posizione globale di ogni volto rilevato utilizzandoFolium, posizionando i volti in base alla latitudine, longitudine e all'angolo.
Infine, il risultato viene registrato in un video di output con sovrapposizioni grafiche, mentre un controllo in tempo reale consente di interrompere il programma con 'q'.
Questa architettura permette di combinare diverse fonti di dati per stimare con precisione le distanze e la posizione globale dei volti, sebbene sia limitata dalla quantità e qualità dei dati di addestramento disponibili.

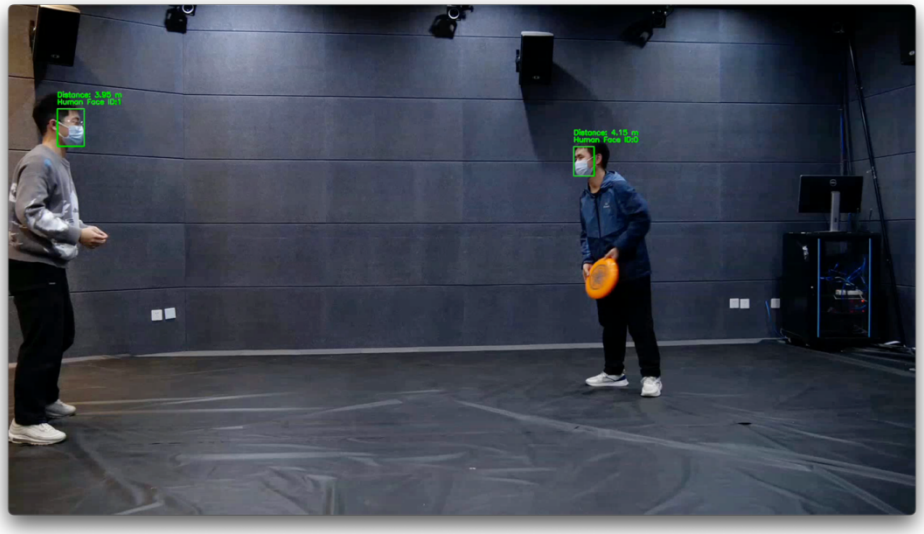
Rete Multimodale

[Descrizione sulla rete multimodale che abbiamo implementato nel progetto]

La rete neurale multimodale MultiModalLSTMModel è progettata per combinare informazioni visive e numeriche al fine di stimare distanze in contesti dinamici.
Utilizza un modello pre-addestrato di **ResNet18** per estrarre le caratteristiche visive da sequenze di frame RGB, producendo un vettore di dimensione 512 per ciascun frame.
Contemporaneamente, i dati numerici, come profondità, latitudine, longitudine e orientamento, vengono elaborati da una rete fullyconnected che riduce la dimensione delle informazioni a 32 feature.
Le caratteristiche visive e numeriche vengono quindi combinate e passate a una **rete LSTM** per catturare il contesto temporale, con l'output della LSTM che rappresenta una sequenza temporale delle informazioni. Infine, un layerfullyconnected è utilizzato per generare la predizione finale della distanza.
Questo approccio consente alla rete di integrare efficacemente le informazioni provenienti da diverse fonti, migliorando la capacità di stima in scenari complessi.

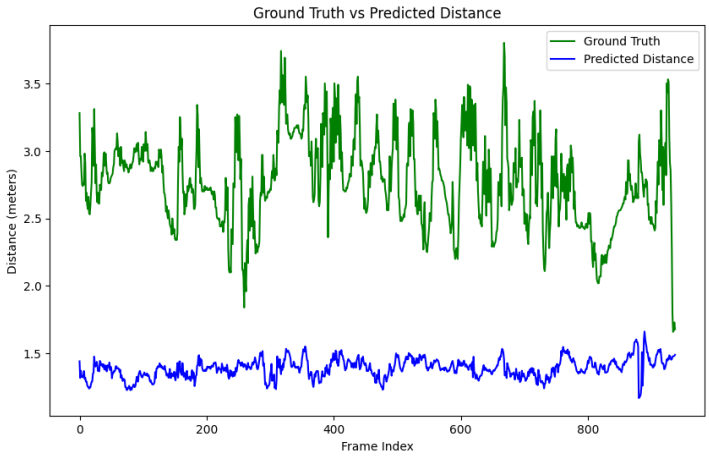
Risultati

[Risultati ottenuti dal progetto]

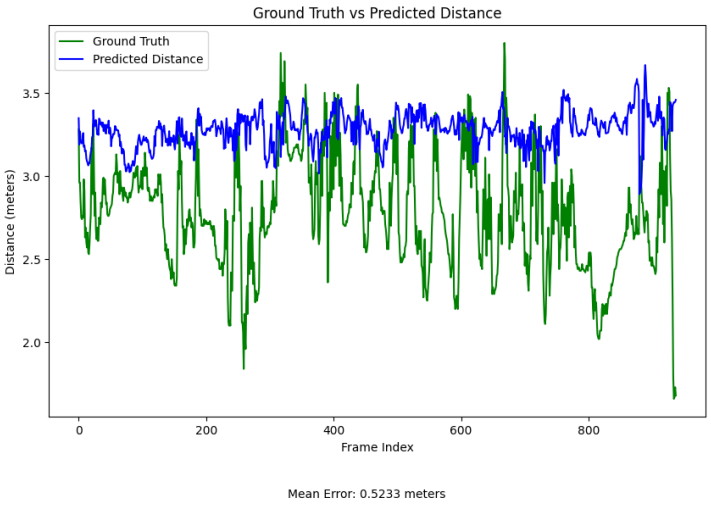


L'immagine mostra l'output ottenuto dal nostro progetto utilizzando il post-processing per questioni di rapidità. Le bounding boxes circondano i volti rilevati indicando l'id del volto per la distinzione e la distanza a cui si trovano gli obiettivi.

Per quanto riguarda l'analisi dei risultati, invece, abbiamo considerato diversi orientamenti della bussola [10.2°,189°,358°].
Qui di seguito ci sono le analisi sull'errore medio.

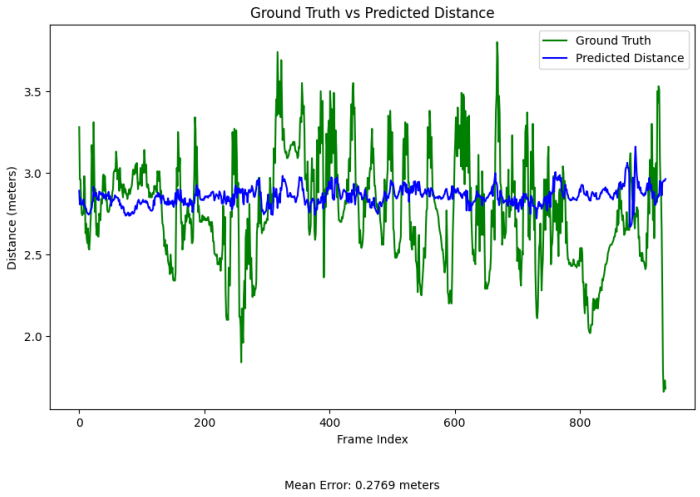


Il primo grafico rappresenta l'analisi del sistema ad un orientamento di 10.5°,in questo caso le predizioni seguono un andamento simile al ground truth.



Il secondo grafico invece rappresenta l'analisi del sistema ad un orientamento maggiore, di 189°. Le predizioni seguono parzialmente il ground truth ma comunque si riesce ad avere un'accuratezza moderata.

Mean Error: 0.5233 meters



Infine, l'ultimo grafico rappresenta l'analisi del sistema ad un orientamento di 358°.
Le predizioni sono significativamente distanti rispetto al ground truth e pertanto si ha una scarsa predizione.

I grafici, mostrano che il sistema si basa su informazioni secondarie (per il nostro sistema stazionario) per effettuare le predizioni sulle varie posizioni.

Problematiche riscontrate

[Problemi riscontrati durante lo sviluppo del progetto e le soluzioni adottate]

Nonostante questo progetto mostri dati promettenti, è ancora lontano dall'essere perfetto.
Le limitazioni del dataset utilizzato per l'addestramento emergono chiaramente durante le predizioni.
Variabili che avremmo voluto far considerare secondarie dalla rete, come il movimento della telecamera, qui volutamente assente poiché la camera è fissa, continuano a influire significativamente sui risultati.
Cambiamenti nell'orientamento o nella posizione della camera all'interno dello stesso file di valutazione producono risultati diversi; tuttavia, i valori rimangono all'interno di un range accettabile.
Le immagini "distance_plot_[0,1,2]" illustrano queste variazioni, evidenziando come i risultati si modifichino all'aumentare dell'angolo di orientamento.
Una possibile soluzione che abbiamo considerato è quella di trattare il sensore come "fisso" per il nostro scopo, permettendo di considerare le diverse angolazioni come contesti distinti.
In questo modo, potremmo addestrare il modello a interpretare ogni angolazione come una variazione predefinita all'interno di uno scenario noto, migliorando la coerenza dei risultati senza dover aumentare eccessivamente il dataset.

Conclusioni

[Conclusioni e possibile sviluppo del progetto]

Questo progetto rappresenta un'estensione del lavoro precedente, arricchito dall'integrazione di dati provenienti da una bussola e da un meccanismo di data fusion implementato tramite una rete neurale.
Sebbene i risultati di valutazione evidenzino la necessità di un dataset più ampio e di un ciclo di addestramento più intensivo, i risultati ottenuti sono promettenti e offrono un'anteprima delle potenzialità del sistema, una volta superate le attuali limitazioni.