Predizione futura di traiettorie: confronto tra Filtro di Kalman e Reti Neurali

Francesco Marchetti
Università degli studi di Firenze
Scuola di ingegneria
francesco.marchetti@stud.unifi.it

1. Introduzione: Problema affrontato

Nell'ambito delle auto a **guida autonoma**, garantire la sicurezza personale e dell'ambiente circostante è la priorità più importante da raggiungere. Uno dei compiti più interessanti da affrontare per configurare il movimento più sicuro per la nostra autovettura è quello di riuscire a predire le **traiettorie future** dei veicoli e dei pedoni che si muovono vicino e accanto a noi. Il futuro lo possiamo definire come una conseguenza di serie di eventi passati, quindi per generare una predizione nel futuro dobbiamo basarci su osservazioni del passato. Nel nostro caso, la traiettoria futura predetta di un veicolo sarà influenzata e determinata dal movimento che ha avuto nel suo passato più prossimo.

Per risolvere questo problema, abbiamo deciso di utilizzare due tipi di approcci. Il primo approccio riguarda lo studio di un modello dinamico basato sulla cinematica utilizzando il **filtro di Kalman**, strumento utilizzato in modo soddisfacente in differenti applicazioni di predizione e di determinazione dello stato di un sistema: la predizione della traiettoria futura sarà generata dalla stima dello stato valutato dalla traiettoria passata. Il secondo approccio si basa su tecniche di **machine learning**, utilizzando varie architetture di **reti neurali artificiali**: sono state addestrare delle reti che, dato come input una traiettoria passata, ci restituiscono la traiettoria futura più probabile.

Gli esperimenti sono stati effettuati sul Dataset **Kitti** [1] contenente traiettorie di tipologie differenti di veicoli e pedoni catturate su scenari e situazioni del **mondo reale**.

2. Dataset

Il Dataset utilizzato in questo lavoro è quello sviluppato da *Karlsruhe Institute of Technology (KIT)* denominato **KITTI**. Da questo dataset abbiamo estratto le traiettorie dei pedoni e delle varie tipologie di veicolo (macchine, biciclette, furgoni e tram) le quali sono state registrate con la frequenza di 10 frame a secondo.

Ogni traiettoria è rappresentata da una sequenza di lunghezza variabile di **punti 2D** espressi in **coordinate mondo**. Negli esperimenti ogni traiettoria completa di un agente è stata **sotto-campionata** tramite *sliding window*, a seconda della lunghezza richiesta dall'esperimento, e abbiamo ipotizzato che ogni traiettoria è **indipendente** dalle altre: ogni predizione futura di un agente si basa solamente sulle osservazioni passate dello stesso agente e della stessa traiettoria sotto-campionata.

Una parte del Dataset Kitti ha disponibile solo i frame acquisiti dalla camera montata nel veicolo adibito alla registrazione. In questo caso le traiettorie non sono più punti 2D in metri in coordinate mondo ma **pixel** nel frame: valutare le prestazioni utilizzando come dati i pixel nell'immagine è difficile non avendo una metrica appropriata. Nel capitolo 5 è stato analizzato un procedimento con cui stimare traiettorie in coordinate espresse in metri partendo da traiettorie definite nel piano immagine tramite **misure reali** della scena e la stima di un'**omografia**.

3. Tecniche

3.1. Filtro di Kalman

Il filtro di Kalman è uno **stimatore ricorsivo** ottimo di tipo *predittore-correttore* che valuta lo stato di un sistema **lineare dinamico** utilizzando una serie di misure osservate nel tempo minimizzando la covarianza dell'errore di stima [2].

Lo **stato** descrive in modo univoco il comportamento dinamico del sistema. Nel nostro problema, lo stato del sistema è rappresentato dalla **posizione** e dalla **velocità** (rispetto l'asse x e y) dell'agente osservato. Oltre lo stato, dobbiamo considerare un'altra variabile, la matrice di **covarianza** dello stato che descrive l'errore della stima corrente.

La **misura** è l'osservazione compiuta sulle variabili dello stato tramite sensori ed è vincolata da una matrice di covarianza che descrive il rumore della stessa.

Il processo ha due fasi: nella prima, fase **predittiva**, generiamo una stima a priori dello stato corrente tramite una **funzione di transizione** che descrive il modello lineare adatto per il problema. Sono stati utilizzati due modelli, quello di **moto rettilineo uniforme** e quello di

moto uniformemente accelerato. Nella seconda fase, fase **correttiva**, aggiorniamo lo stato incorporando la **misura** osservata per ottenere una stima a posteriori migliore. L'aggiornamento viene controllato dal *guadagno di Kalman* che definisce quanto la nuova misura deve essere usata per aggiornare la stima a priori. Il guadagno di Kalman dipende dalla covarianza dell'errore di stima e dalla covarianza dell'errore della misura: più il guadagno è alto e più la misura avrà peso per l'aggiornamento della stima.

Il processo si itera per tutta la lunghezza della traiettoria passata osservata per calcolare lo stato del movimento dell'agente: le misure usate per aggiornare lo stato sono le posizioni reali a ogni istante di tempo. Ricavata la stima dello stato, la traiettoria futura viene generata propagando via via lo stato nel passo di predizione senza passare nella fase di correzione, iterando la propagazione a seconda della lunghezza della traiettoria futura desiderata.

3.2. Reti Neurali Feed-forward

Le **reti neurali** sono una tecnica di **machine learning** che si basano sulla costruzione di un modello matematico che simuli la struttura e il comportamento del cervello umano. La rete è composta da neuroni artificiali interconnessi tra loro i quali, dato un certo input, estraggono informazioni che vengono propagate ai neuroni successivi fino a restituire il valore di un output desiderato. Ogni neurone è composta da **pesi** e **'bias'** che gestiscono la propagazione dell'informazione.

Le reti neurali sono organizzate in *layer*: in questo lavoro abbiamo utilizzato le reti **feed-forward** in cui le connessioni collegano i neuroni di un layer con quelli del layer successivo, perciò l'informazione viene propagata in una sola direzione.

Le strutture che sono state usate per gli esperimenti sono:

- **Single-layer feed-forward** (lineare): l'input della rete è collegato a un singolo layer di neuroni con il compito di effettuare una **trasformazione lineare** che genera direttamente la traiettoria futura.
- **Multi-layer feed-foward** (non-lineare): la rete è composta da 2 layers di neuroni, intervallate da una **funzione di attivazione** denominata *ReLU*, $f(x) = \max(0, x)$, per introdurre la proprietà di non linearità.

La rete neurale viene addestrata tramite l'algoritmo di retropropagazione dell'errore: i pesi e i bias di ogni neurone vengono modificati per fare avvicinare l'output della rete con l'output desiderato.

4. Esperimenti

In tutti gli esperimenti effettuati, la lunghezza della traiettoria futura da predire è stata fissata a **4 secondi**. La dimensione del test set è di **1600 traiettorie**. Per valutare le prestazioni sono state utilizzate le seguenti **metriche** (in metri) tra la predizione calcolata dai vari metodi e la traiettoria corretta:

- **media** della distanza euclidea L2 tra tutti i punti della traiettoria.
- distanza euclidea L2 a differenti **istanti temporali** (1, 2, 3, 4 secondi).
- media della **distanza di Hausdorff modificata** [3] tra tutti i punti della traiettoria.

La distanza di Hausdorff modificata compara le due traiettorie confrontando ogni punto della traiettoria predetta con il punto più vicino ad esso della traiettoria corretta. Mentre la distanza euclidea penalizza disallineamenti temporali e spaziali, la distanza di Hausdorff modificata penalizza solo la **divergenza spaziale** senza punire divergenze temporali.

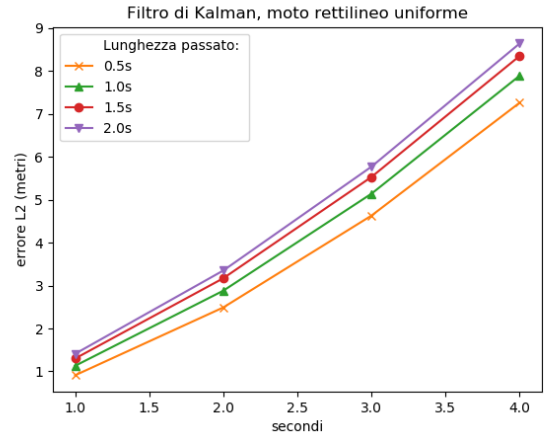
Come primo esperimento sono state analizzate le prestazioni del filtro di Kalman dati i due diversi **modelli di moto** e differenziando la **lunghezza** della traiettoria **passata** utilizzata per stimare lo stato del filtro. I risultati sono riportati in figura 1. Come possiamo osservare, più corta è la sequenza usata per stimare lo stato e minore è l'errore in termini di distanza tra la predizione e la traiettoria corretta. Perciò, per utilizzare il filtro di Kalman in questo problema è sufficiente una traiettoria passata di breve lunghezza. Un esempio qualitativo è riportato nelle figure 3a, 3b, 4a, 4b

In seguito è stato fatto lo stesso esperimento utilizzando le architetture di reti neurali descritte precedentemente con dimensione del passato di 2 secondi. Le reti neurali sono state addestrate con 5000 esempi di traiettorie, ottimizzatore Adam con learning rate di 0.0001 per 600 epoche, dimensione del batch di 32. In tabella 1 sono riportati i migliori risultati ricavati dagli esperimenti effettuati con il filtro di Kalman in entrambi i modelli e i risultati generati dalle reti neurali. Dalla tabella si evince che, mentre nell'istante temporale a 1 secondo la variazione tra gli errori dei vari metodi è bassa, negli istanti temporali maggiori la varianza tra essi aumenta a vantaggio delle reti neurali, le quali hanno migliori prestazioni. Questo si riflette sui risultati medi sulla metrica L2 e MHD. Un esempio qualitativo sull'uso delle reti neurali è riportato nelle figure 5a, 5b.

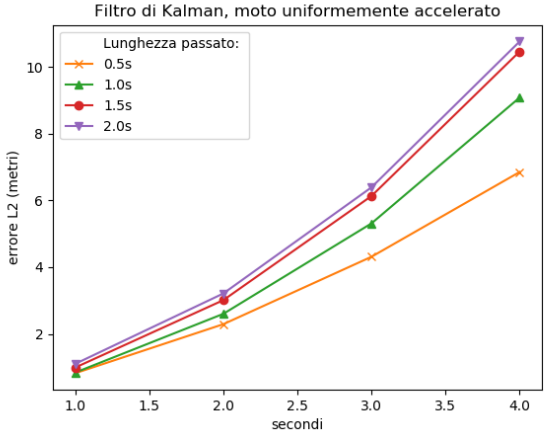
5. Traiettorie nel piano immagine

5.1. Problema

Una porzione del dataset Kitti ha disponibile solamente le traiettorie definite nel piano immagine anzichè in coordinate 2D mondo espresse in metri e i punti della traiettoria sono situati nel **piano** dove si muove l'agente(ad esempio un pedone si muove in un piano di un marciapiede). Se



(a) modello moto rettilineo uniforme.



(b) modello moto uniformemente accelerato.

Figure 1: Prestazione Filtro di Kalman al variare della dimensione del passato considerato

	Errore (metri)					
	istanti temporali				media	
Metodi	1.0s	2.0s	3.0s	4.0s	L2	MHD
KF rett. unif.	0.91	2.49	4.63	7.26	2.95	1.92
KF unif. acc.	0.82	2.29	4.31	6.85	2.75	1.81
FF single-layer	0.82	2.10	3.77	5.65	2.3	**1.23**
FF multi-layer	**0.79**	**2.01**	**3.56**	**5.24**	**2.18**	1.31

Table 1: Prestazione dei metodi utilizzati. I risultati riportati sono le medie rispetto al dataset di test.

utilizzassimo uno dei metodi proposti per generare la predizione, questa sarebbe espressa ancora in coordinate pixel e non sarebbero utilizzabili le metriche definite in questo lavoro.

Dato un frame contenente degli agenti in movimento con la traiettoria passata e futura (figura 6a), se conoscessimo delle misure reali del piano contenente la traiettoria da predire, potremo mappare i pixel dell'immagine in coordinate del mondo reale tramite un'omografia planare.

5.2. Omografia Planare

L'omografia planare è una **trasformazione 2D-2D** che mappa punti di due piani diversi tale che ad ogni punto del primo piano corrisponde uno e un solo punto del secondo piano[4]. Tale trasformazione si esprime matematicamente tramite il prodotto dei punti per una matrice $\boldsymbol{H}$ 3x3 tale che

$$x_i' \sim \boldsymbol{H} x_i, \forall i \tag{1}$$

con x_i e x_i' insieme dei punti rispettivamente del primo e del secondo piano. H è una matrice

$$\boldsymbol{H} = \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{pmatrix} \tag{2}$$

con $det(\boldsymbol{H}) \neq 0$ quindi **invertibile** e definita a meno di un fattore di scala con 8 gradi di libertà: per fissare i gradi di libertà e quindi calcolare l'intera omografia servono almeno **4 corrispondenze** di punti.

5.3. Realizzazione

Conoscendo almeno **4 distanze**, e conseguentemente **4 punti**, del piano nel mondo reale, possiamo definire un nostro sistema di riferimento espresso in metri: uno dei punti sarà l'origine mentre gli altri punti saranno definiti dalla distanza da esso. Associando a ogni punto trovato precedentemente il pixel corrispondente nel frame, possiamo stimare un'omografia che mappi, in seguito, tutti i pixel della traiettoria nel nostro sistema di riferimento.

In questo modo, nel nuovo sistema di riferimento possiamo generare una predizione del futuro tramite la traiettoria passata e in seguito possiamo valutare la predizione confrontandola con la traiettoria futura corretta utilizzando una delle metriche definite precedentemente.

In figura 6a siamo riusciti a ricavare le misure reali del rettangolo evidenziato: conoscendo i pixel corrispondenti agli angoli del rettangolo abbiamo calcolato l'omografia. Sempre in figura 6a abbiamo un frame dove sono rappresentate la traiettoria passata e futura di un agente (in questo caso un pedone). Con l'omografia appena calcolata, possiamo mappare le traiettorie nel sistema di riferimento mondo espresso in metri , generare la predizione (figura 6b) e valutarla con una metrica a nostra scelta, ad esempio una distanza euclidea in metri. Infine, possiamo mappare la predizione generata nel frame tramite l'omografia **inversa** (figura 6c): la traiettoria predetta sarà ora nel piano immagine

5.4. Analisi

Prendiamo una singola sequenza di registrazione del dataset in cui sono disponibili sia le traiettorie descritte sul piano immagine sia sul mondo reale e osserviamo la differenza tra quest'ultima e quella generata in coordinate mondo tramite il procedimento appena descritto.

Per ogni frame della sequenza, mappiamo la traiettoria di ogni agente nel sistema di riferimento definito dall'omografia (figura 7a e 7b). Prima di poterle confrontare, però, è necessaria un'altra trasformazione poichè la traiettoria reale e quella generata sono in sistemi metrici **differenti** tra loro(figura 7c). Per ovviare a questo inconveniente, **trasliamo** le traiettorie in modo tale che il punto rappresentante l'istante presente si trovi nell'**origine** e stimiamo una funzione di **rotazione** per allinearle (figura 7d).

La metrica utilizzata è la media della distanza euclidea L2 punto a punto tra la traiettoria generata e quella corretta. In figura 2 è riportato l'istogramma che mostra la frequenza dell'errore secondo tutte le traiettorie della sequenza considerata. I risultati trovati sono soggetti principalmente dalla stima dell'omografia, dal comportamento e dalla posizione della traiettoria nel frame: trovando corrispondenze migliori tra il piano immagine e il piano del mondo possiamo ridurre la distanza tra la traiettoria generata e quella corretta ma le traiettorie di agenti lontani rispetto alla camera saranno comunque mappate in modo rumorose nel piano mondo.

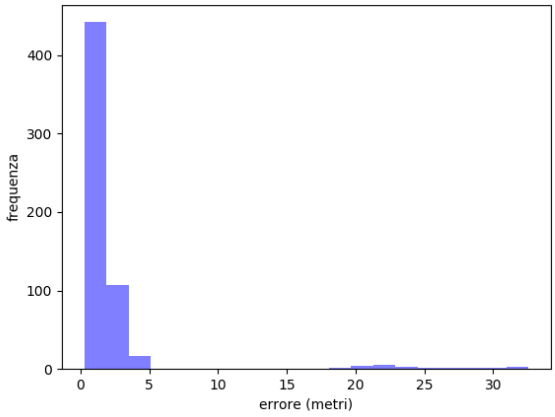
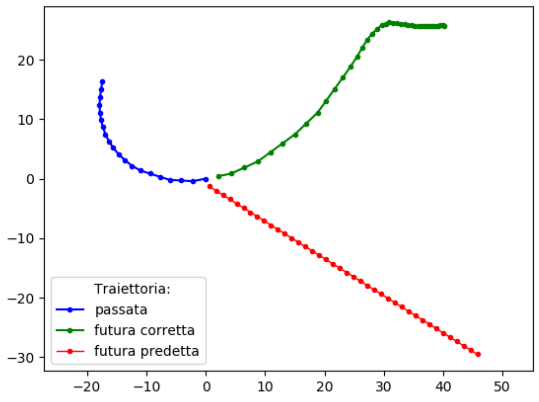


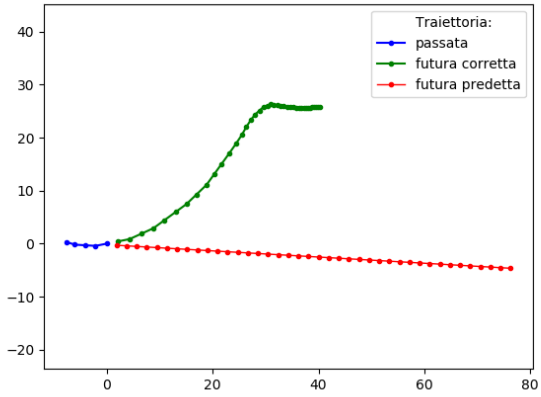
Figure 2: istogramma degli errori su 587 traiettorie di una singola sequenza di registrazione in una scena fissata.

6. Conclusione

In questo lavoro è stato analizzato il comportamento di due tipologie di tecniche che sono appropriate per affrontare il problema di predizione di traiettorie di veicoli e pedoni. Abbiamo dimostrato che con l'utilizzo di una tecnica di machine learning, reti neurali feed-forward, siamo riusciti a raggiungere prestazioni migliori rispetto all'uso di un filtro di Kalman.



(a) dimensione del passato: 20 punti



(b) dimensione del passato: 5 punti

Figure 3: Predizione utilizzando un filtro di Kalman con moto rettilineo uniforme

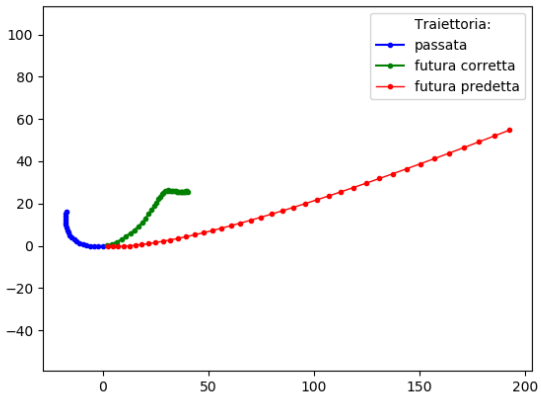
Inoltre, è stato analizzato un procedimento per utilizzare le tecniche sviluppate in questo lavoro quando abbiamo a disposizione solo traiettorie definite sul piano immagine.

References

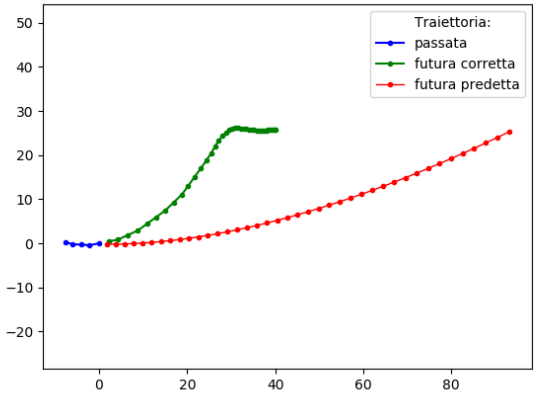
[1] Geiger, A., Lenz, P., Stiller, C., and Urtasun, R. (2013). Vision meets robotics: The KITTI dataset. The International Journal of Robotics Research, 32(11), 1231–1237.

[2] Cuevas, Erik, Zaldivar, Daniel, Rojas, Raul. (2005). Kalman Filter for vision tracking. Measurement. 33.

[3] M.-P. Dubuisson and A. K. Jain, "A modified hausdorff distance for object matching," in Pattern Recogni-



(a) dimensione del passato: 20 punti

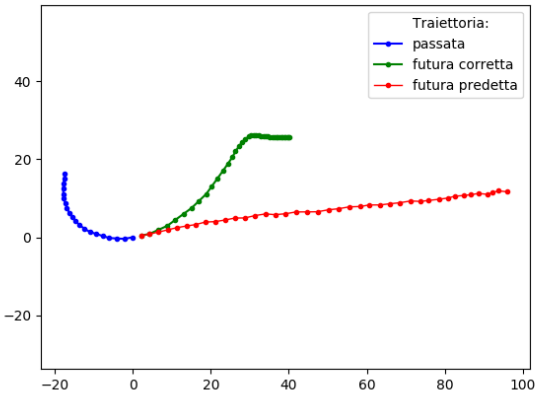


(b) dimensione del passato: 5 punti

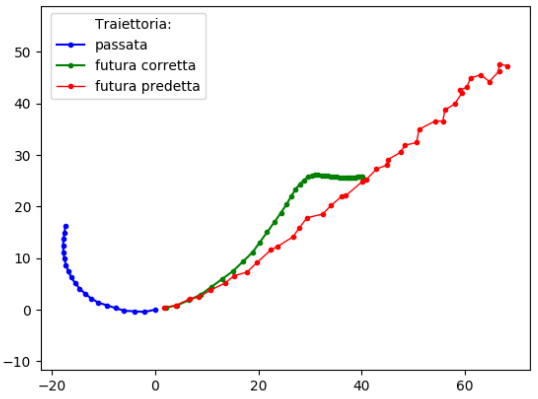
Figure 4: Predizione utilizzando un filtro di Kalman con moto uniformemente accelerato

tion, 1994. Vol. 1-Conference A: Computer Vision Image Processing., Proceedings of the 12th IAPR International Conference on, vol. 1. IEEE, 1994, pp. 566–568.

[4] Richard Hartley and Andrew Zisserman, *Multiple View Geometry in Computer Vision*, Second Edition, Cambridge University Press, 2004.

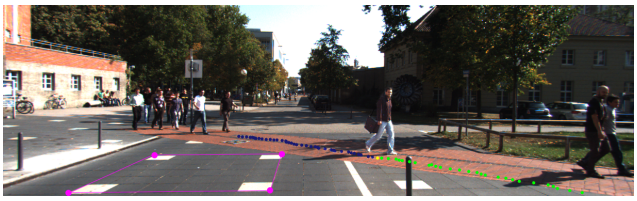


(a) Predizione con Rete neurale single-layer

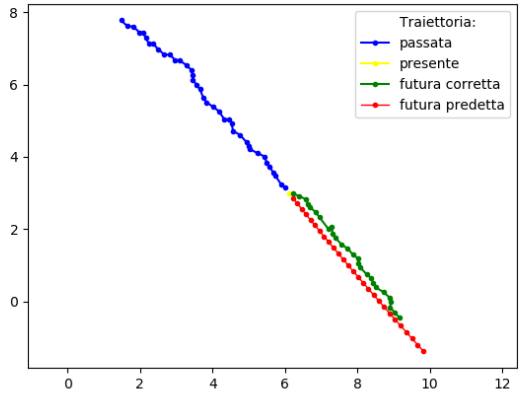


(b) Predizione con Rete neurale multi-layer

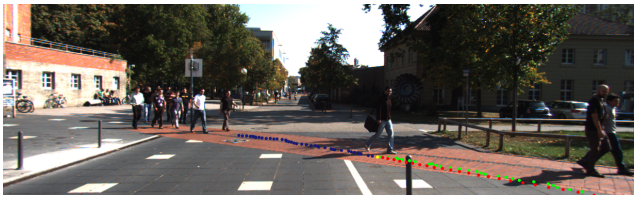
Figure 5: Predizione utilizzando una rete neurale



(a) Frame con agenti in movimento: la traiettoria passata (in blu) e la traiettoria futura (in verde) sono riferite al pedone evidenziato. Il rettangolo (in viola) è utilizzato per la stima dell'omografia.

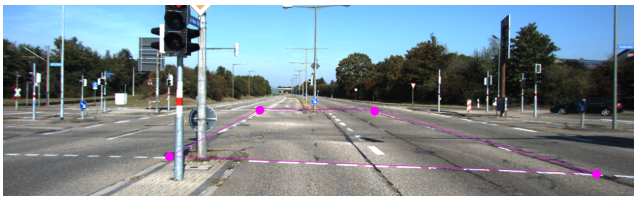


(b) traiettoria nel sistema di riferimento mondo espresso in metri.



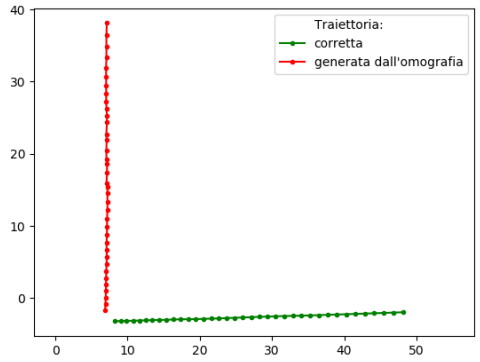
(c) Predizione (in rosso) proiettata nel piano immagine con l'omografia inversa

Figure 6: Procedimento per mappare traiettorie nel piano mondo, generare una predizione del futuro e mapparla nel piano immagine.

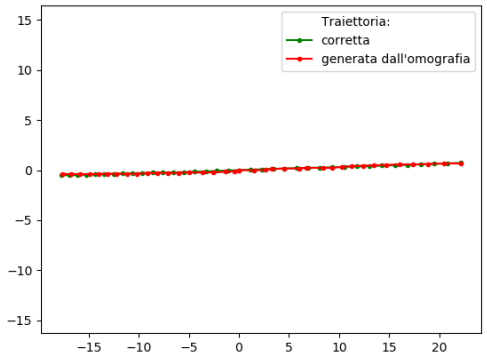


(a) Scena della sequenza da cui calcolare l'omografia (tramite il rettangolo evidenziato in viola).

(b) Esempio di frame contenente una traiettoria



(c) traiettorie in differenti sistemi di riferimento



(d) traiettorie traslate e ruotate nello stesso sistema di riferimento

Figure 7: Procedimento per confrontare la traiettoria corretta nel piano mondo e quella generata dal piano immagine tramite l'omografia