Relazione progetto di Image and Video Analysis

Leonardo Casini
Andrea Simioni
Università degli studi di Firenze

December 4, 2018

Abstract

Il progetto si pone l'obiettivo di confrontare l'accuratezza del tracking di oggetti 3D osservati in ambito automotive, nel quale l'associazione dei dati è operata mediante overlapping e matching in frame adiacenti. Le immagini analizzate provengono da una singola camera, mentre i metadati sono estratti da un sistema GPS montati su un' auto. Per quanto riguarda la detection, sono state utilizzate due o più diverse CNN.

1 Introduzione

Ad oggi un sistema di guida autonoma evoluto è in grado di accorgersi dell'imminente impatto con un oggetto ed in conseguenza a ciò mette in atto una azione di frenata.
Lo scenario che si immagina per il futuro è diverso: un sistema non solo dovrà essere in grado di accorgersi del rischio di impatto, ma dovrà effettuare una predizione sulla traiettoria futura di ciascun "oggetto" osservato ed in base a questa pianificare una traiettoria di sicurezza che consenta di evitare la collisione non semplicemente frenando, ma anche attraverso un cambio di traiettoria.
Osservando una scena con una camera calibrata posta su una macchina e sfruttando delle assunzioni sul moto degli oggetti nel piano stradale è possibile effettuare una detection degli oggetti nella scena e ricostruirne la posizione in un sistema 3D. Una volta che saranno ottenute grandi quantità di traiettorie ricostruite per classe di oggetti sarà possibile stimare, attraverso

vari approcci (Machine Learning e non), la traiettoria futura di altri oggetti delle medesime classi.

2 Metodi e Obiettivi

Per lo sviluppo del progetto, è stato utilizzato il dataset messo a disposizione dal Karlsruhe Institute of Technology (KIT) per il KITTI 1 Vision Benchmark .
Il dataset è composto da immagini catturate guidando attraverso la città di medie dimensioni Karlsruhe, in aree rurali e in autostrada. In ogni immagine sono visibili fino a 15 auto e 30 pedoni. Le immagini ed i relativi metadati vengono raccolti attraverso un' auto station-wagon equipaggiata con diverse videocamere (due a colori e una in bianco e nero), un sistema di localizzazione GPS ed uno scanner laser Velodyne.
L' obiettivo del progetto è quello di verificare il livello di precisione nella ricostruzione della traiettoria degli oggetti osservati utilizzando immagini provenienti da una singola camera e una detection basata su tre diverse CNN. Si vogliono esprimere le traiettorie degli oggetti nel sistema di riferimento al fine di seguire con la miglior accuratezza possibile l'andamento nel flusso ottico. Si effettua quindi la detection utilizzando delle Reti Neurali Convolutive che ci forniscono una stima della posizione degli oggetti all'interno del frame ed una loro classificazione. Attraverso queste informazioni è possibile quindi generare le rispettive tracce di ogni oggetto nel flusso ottico.

3 Metodi e Obiettivi

Il codice sviluppato è composto dai seguenti moduli:

- Estrazione informazioni dal dataset
- Tracking degli oggetti nel flusso ottico basato su overlap e matching
- Stima della correttezza delle detection

3.1 Estrazione Informazioni dal dataset

Il dataset è composto da diversi gruppi di immagini in successione, ognuno indipendente dall'altro, catturati durante un breve tragitto. Ogni gruppo può rappresentare diverse condizioni stradali come: città, campagna o autostrada; trafficate e non. Per ogni set di immagini sono state forniti:

- 'img' → BGR frame
- 'annot' → ground truth
- 'dets' → Mask-RCNN detections
- 'masks' → masks ricavate da Mask-RCNN
- 'feats' → Features Vectors

4 Software sviluppato

Il Software è stato sviluppato in Python(Versione 3.x) utilizzando come base di partenza per l'implementazione le repositories di *github* situate in fede-becat/kitti_playground. Le dipendenze del progetto sono i seguenti moduli Python:

- openCV
- nunpy
- pickle
- motmetrics

4.1 Tracking

Vengono realizzate tre implementazioni di Tracking degli oggetti in base a 'dets', 'mask' e 'feats' ognuna delle quali associa un serie di elementi ad un oggetto in modo da poter essere rilevato.
Durante la detection, per ciascun oggetto tramite 'dets' vengono rilevati nel flusso ottico i seguenti dati: *box*, *obj_type*, *score* vedi fig. 1.
Il *box* è il BoundingBox dell'oggetto ovvero le coordinate del bordo rettango-

Figure 1: Rappresentazione dei dati del dets

lare che racchiude completamente l'oggetto. L'*obj_type* invece è una stringa che indica la classe di appartenenza dell'oggetto; serve a filtrare macchine da pedoni o biciclette. Infine lo *score* è un'attributo di tipo float che da una stima sull'accuratezza di detection del *bounding boxes* e serve a filtrare le migliori predizioni.
Infatti poiché per ogni oggetto è associato più di un *bounding boxes*, è stato effettuato un filtraggio in base allo score mediante la funzione *Non Maximal Suppression*(vedi fig. 2), la quale restituisce un insieme di indici relativi ai box migliori.

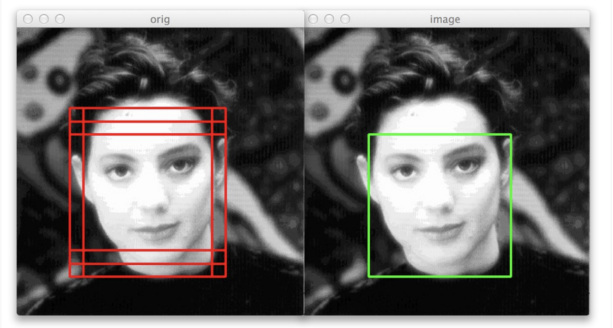

Figure 2: Non Maximal Suppression

Per realizzare il tracking degli oggetti rilevati, si definisce in prima istanza una classe di nome Track avente i seguenti attributi:

- **ID** = numero intero identificativo univoco dela traccia
- **boxes** = array contenente i boxes salvarti in ordine di frame associati all'oggetto di interesse
- **sequence** = array contenente gli indici dei boxes
- **color** = colore identificativo il cui scopo è la visualizzazione grafica della traccia nel flusso ottico
- **state** = stato corrente della traccia
- **startingFrame** = frame inizio traccia

Il metodo di tracking utilizzato per i bounding box sfrutta il principio di Intersection Over Union fra frame adiacenti. Nell' Intersection Over Union viene calcolato il rapporto tra l'intersezione e l'unione delle aree di due bounding boxes(il calcolo delle aree avviene sfruttando le coordinate dei vertici dei box vedi fig. 3).

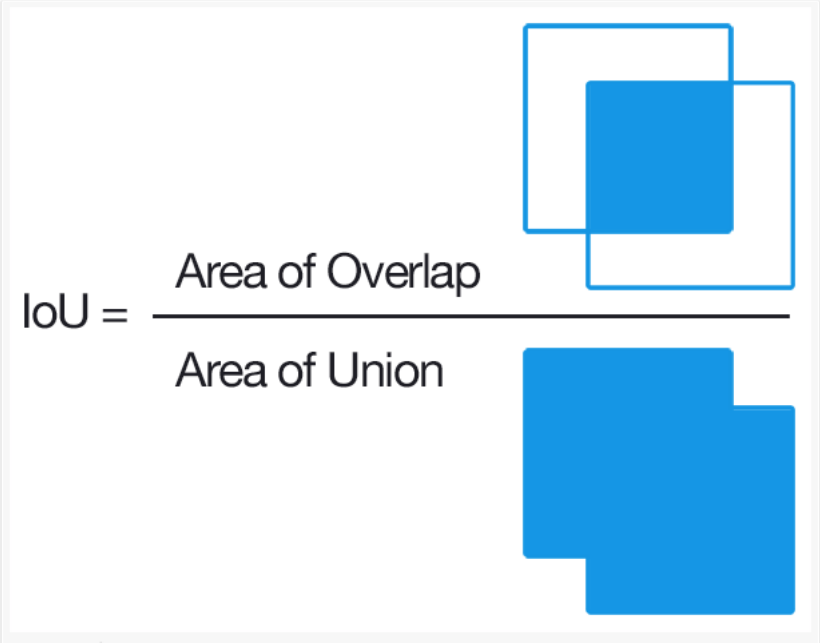

Figure 3: Intersection over Union

Vengono dunque inizializzati due array di supporto *currentBoxes* e *pastBoxes* per permettere il calcolo di tale rapporto fra boxes di frame adiacenti. Per ogni coppia di box viene calcolato il rapporto IoU e salvato in una matrice nella quale l'indice di riga rappresenta un elemento dell'array *pastBoxes* mentre l'indice di colonna rappresenta l'elemento di *currentBoxes* . La matrice ottenuta confrontando i boxes del frame 0 con il frame 1 serve alla inizializzazione dei primi oggetti di classe Track.
Viene avviato un ciclo in cui ad ogni iterazione vengono selezionati il valore massimo della matrice e i suoi relativi indici di riga(r) e colonna(c). Nel caso il valore massimo sia maggiore del parametro *Iou_Treshold* viene creata una Track in cui viene salvato r-esimo elemento dei *pastBoxes* e il c-esimo elemento dei *currentBoxes*. L'attributo state viene quindi inizialmente settato a 'new'. Successivamente viene azzerata la r-esima riga e la c-esima colonna in quanto quei due boxes sono già stati associati: il ciclo termina quando tutti gli elementi della matrice sono nulli.
Nelle successive creazioni della matrice, al fine di poter tracciare in modo continuativo gli oggetti presenti nel flusso ottico e di limitare la creazione di tracce ridondanti, mediante il metodo findTracks si esegue sempre un' operazione di Intersection Over Union tra il box selezionato presente come elemento di riga nella matrice e di tutti gli ultimi boxes salvati nelle tracce già inizializzate. In caso di esito positivo (risultato maggiore dell 70 %), il box selezionato era già contenuto in una traccia; Il salvataggio del secondo box con cui è avvenuto il matching quindi avviene su quest' ultima. L' attributo state delle tracce aggiornate passa quindi da 'new' ad 'active'.

Al contrario, se non vi è alcun riscontro, vi è l'inizializzazione di una nuova traccia nel quale i primi due box salvati al suo interno corrispondono ai box selezionati. Ad ogni confronto di boxes adiacenti viene inizializzata una lista in cui vengono salvati gli *id* delle tracce che sono state aggiornate, in modo che quelle che non hanno avuto alcun riscontro vengano cancellate tramite il metodo *delete* della classe Track. Tale metodo incrementa un contatore denominato countNoMatch e setta lo stato della tracce a "dying". Nel caso esso superi il valore di soglia impostato, lo stesso metodo imposta lo stato a 'dead, "uccidendo" la traccia in modo tale da non renderla più attiva. I risultati quindi ottenuti sono stati salvati in un database in modo tale da poter effettuare successivamente una fase di testing della qualità del tracking.

4.2 Tracking delle mask e features

Un procedimento analogo viene utilizzato anche per le masks e i features vectors, con alcune variazioni nel metodo del confronto. Gli indici calcolati nella

Non Maximal Suppression vengono utilizzati analogamente per il filtraggio. Per quanto riguarda le maks, non avendo infatti più i bounding box ma la forma degli oggetti nel flusso, l'Intersection Over Union descritto precedentemente non risulta più efficace. Viene quindi introdotta una variante logica di tale metodo, dove l intersezione è ottenuta mediante un AND logico, e l'unione con un OR. Per poter utilizzare ciò, le masks sono convertite in formato Booleano. Gli oggetti di tipo Track inizializzati contengono questa volta la sequenza di masks in cui si è verificato il matching.
Per quanto riguarda invece i features vectors, il metodo di matching è applicato mediante la *cosine similarity*. Tale misura calcola il coseno tra i due vettori tramite la formula:

$$similarity = cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|} \tag{1}$$

In base alla definizione del coseno, dati due vettori si otterrà sempre un valore di similitudine compreso tra -1 e +1, dove -1 indica una corrispondenza esatta ma opposta (ossia un vettore contiene l'opposto dei valori presenti nell'altro) e +1 indica due vettori uguali. Poiché tuttavia si lavora in uno spazio positivo, i risultati ottenuti sono sempre delimitati dall'intervallo [0,1]. I features vectors con cui è avvenuto il match sono salvati in oggetto di tipo Traccia, come è avvenuto in modo analogo per le 'dets' e le 'masks'.

5 Testing

Al fine di verificare la qualità del tracking è stata implementata una successiva fase di testing. Lo script presenta tre diversi parametri, la cui variazione comporta analogamente risultati differenti, In particolare:

- *IouTreshold* = Stabilisce la soglia minima necessaria per avere un valore di Intersection Over Union sufficientemente positivo ai fini dell'analisi. I Boxes, il cui Iou risulta essere inferiore alla soglia, non sono né inseriti in tracce preesistenti, né danno luogo alla creazione di nuove di esse.

- *ScoretreshHold* = Parametro necessario al filtraggio dei boxes in base allo score. I Boxes il cui score è inferiore allo Scoretreshold sono scartati.

- *countNoMatch* = Parametro che definisce il numero massimo di volte in cui una traccia può non essere aggiornata durante lo scorrere dei frames. Superato questo valore, la traccia passa dallo stato "Active" allo stato "Death", e quindi non risulta essere più attiva.

La libreria utilizzata come metrica di analisi è Py-motmetrics, reperibile tramite il link *github* situate in cheind _py-motmetrics (Si può comunque installare tramite pip3). Py-motmetrics fornisce un'implementazione Python che permette di valutare in modo quantitativo il multitracking sviluppato in base a diversi parametri, tra cui:

- MOTA (Multiple Object Tracking Accuracy) = Una delle metriche più utilizzate per valutare le prestazioni di un tracker, definita mediante l'espressione:

$$MOTA : 1 - \frac{\sum_t (FN_t + FP_t + IDSW_t)}{\sum_t GT} \tag{2}$$

  dove t è l'indice del frame corrente, GT il numero di oggetti ground truth, FP il numero di falsi positivi(oggetti rilevati erroneamente dal tracker), FN il numero di falsi negativi (oggetti non rilevati dal tracker ma presenti nel flusso ottico) mentre IDSW (identity switch) è l'errore di mismatch, il quale viene conteggiato se un target di ground truth i è abbinato alla traccia j e l'ultimo assegnamento noto è k $\neq$ j. MOTA può anche essere negativa nei casi in cui il numero di errori commessi dal tracker superi il numero di tutti gli oggetti nella scena.

- MOTP (Multiple Object Tracking Precision): indica la dissomiglianza media tra tutti i veri positivi e i corrispondenti groundtruth targets. È definita mediante:

$$MOTP = \frac{\sum_{t,i} d_{t,i}}{\sum_t c_t} \tag{3}$$

  dove c_t denota il numero di match durante il frame t e d_t,i è l'overlap tra il bounding box del target i con il corrispettivo ground truth assegnato.

- Tre misure di qualità delle tracce: Ogni traiettoria di groundtruth può essere classificata come:
  - Mostly Tracked (MT): se il target specifico è tracciato per almeno l'80% durante il suo periodo di vita nel flusso ottico
  - Mostly Lost (ML): se il target specifico è tracciato per meno del 20% del suo periodo di vita
  - Partially Tracked(PT): tutte le tracce rimanenti.

- Recall: indica il rapporto tra tutti i veri positivi e il numero totale di elementi che appartengono effettivamente alla classe positiva (cioè la somma di veri positivi e falsi negativi).

- Num_fragmentations definito come:

$$Num\_fragmentations = \frac{FM}{Recall} \tag{4}$$

  dove FM indica la frammentazione delle tracce (conta quante volte si è interrotto il tracciamento delle traiettorie di groundtruth)

- Precision: definita come il numero di oggetti rilevati in base alla somma dei falsi positivi e dei rilevati stessi.

Come prima cosa, si inizializza l'accumulatore mediante la funzione:

$$acc = mm.MOTAccumulator(auto_id = True) \tag{5}$$

L' accumulatore viene via via popolato allo scorrere dei frame mediante la funzione update, la quale prende in ingresso gli identificativi dei groundtruth e delle tracce e la matrice delle distanze (calcolata in modo analogo alla matrice di Intersection Over Union descritta precedentemente). Mediante la funzione create si inizializzano le metriche; Il calcolo viene poi eseguito mediante la funzione compute. I risultati quindi sono salvati in file csv in modo tale che l'utente possa usufruirne facilmente mediante l'uso di tabelle.
I test sono stati effettuati su due video,'0000' e '0004', rispettivamente di 153 e 313 frames. E' risultato che all'aumentare del countNoMatch (fino al valore massimo di 4), a parità di Iou_Treshold e Score_Treshold, i valori di MOTA ,MOTP, Precision e Recall aumentano. Sono stati effettuati ulteriori test variando sia l' Iou_Treshold che Score_Treshold mantenendo fisso il countNoMatch a 4, da cui è risultato:

- per la iou_Treshold, i valori ottimali sono ottenuti quando questa è compresa tra 0.3 e 0.5 in quanto a valori alti di questa soglia vi è un aumento della precision, però un'eccessiva diminuzione degli oggetti rilevati nel flusso ottico, e quindi una successiva diminuzione dei valori di MOTA e MOTP. A valori bassi invece l' aumento degli rilevati comporta un conseguente aumento dei valori di MOTA e MOTP a scapito dei valori di precision.

- per la Score_Treshold, i valori ottimali sono ottenuti sempre per valori intermedi come nel caso dell'Iou_Treshold, in quanto a valori alti si ha un'eccessivo filtraggio dei predicted boxes, comportando un aumento della precision ma una diminuzione di MOTA e MOTP. Per valori bassi un eccessivo numero di predicted boxes comporta invece un'eccessiva diminuzione della precision.

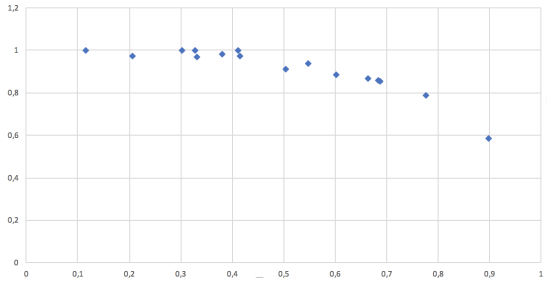

Figure 4: Grafico Precision Recall

Il Parametro Recall è risultato inversamente proporzionale rispetto alla Precision, seguendo un andamento descritto in figura 4 (Nelle ascissa si ha la recalll mentre nell'ordinata la Precision) .

In generale quindi i risultati migliori risultano essere a valori intermedi delle due soglie descritte precedentemente. l'ottimo per quanto riguarda la MOTA (a circa 0,45) è stato raggiunto a 0.2 di Iou_Treshold e 0.3 di Score_Treshold, mentre per la MOTP (circa 0,28) a 0,1 e 0,1 rispettivamente. Per il valore della Precision si ottiene l' ottimo (che corrisponde al valore unitario) con più casi (ad esempio con valori di 0.4 e 0.8) In figura 5 si possono consultare tutti i risultati con countNoMatch settato a 4 del video '0004'(Per ulteriori valori con countNoMatch diversi, consultare la tabella riportata in Multitracking Tracking).

4	0,1	0,1	313	0,14806298	0,28175308	24	103	576	92	23	6	1	0,585313175	0,898342541
4	0,1	0,3	313	0,451933702	0,24177964	66	107	106	283	11	18	1	0,854395604	0,687292818
4	0,2	0,2	313	0,449723757	0,279755955	54	111	185	202	22	7	1	0,791666667	0,77679558
4	0,2	0,3	313	0,453038674	0,236688742	69	108	100	287	11	18	1	0,860724234	0,682872928
4	0,3	0,3	313	0,444198895	0,237427461	68	108	91	304	8	20	2	0,86849711	0,664088398
4	0,4	0,3	313	0,411049724	0,253812166	69	105	68	360	4	21	5	0,889070147	0,602209945
4	0,4	0,4	313	0,419889503	0,237374291	66	84	32	409	2	20	8	0,939393939	0,548066298
4	0,4	0,8	313	0,366850829	0,119595663	54	41	0	532	1	15	14	1	0,412154696
4	0,5	0,3	313	0,370165746	0,278492609	61	79	42	449	3	14	13	0,915662651	0,503867403
4	0,5	0,5	313	0,342541436	0,176400287	56	55	10	530	1	15	14	0,974025974	0,414364641
4	0,5	0,6	313	0,323756906	0,119294099	50	45	6	561	1	11	18	0,982857143	0,380110497
4	0,5	0,8	313	0,295027624	0,094689003	43	30	0	608	1	8	21	1	0,328176796
4	0,5	0,9	313	0,276243094	0,065403049	36	24	0	631	1	9	20	1	0,302762431
4	0,6	0,5	313	0,278453039	0,15159362	40	39	9	605	0	8	22	0,970873786	0,331491713
4	0,8	0,5	313	0,17679558	0,088525938	27	21	5	719	0	4	26	0,97382199	0,205524862
4	0,9	0,5	313	0,098342541	0,033088061	19	16	0	800	0	1	29	1	0,116022099

Figure 5: Risultati con vari parametri per il video 0004

6 Sviluppi Futuri

Possibili sviluppi futuri riguardano il calcolo dell'accuratezza delle tracce di masks and features, poiché la libreria Motmetrics risulta essere vincolata al concetto di BoundingBox e quindi è risultata adoperabile solo per le detections. Un iniziale approccio è stato sviluppato salvando la sequenza di indici relativi agli oggetti nel frame per ogni implementazione. Dato che dagli studi effettuati sulle tracce l'i-esimo oggetto è individuato dall' i-esimo box, mask e features vector, un possibile test di accuratezza è confrontare se le sequenze corrispondono (o al massimo differiscono di pochi elementi). Inoltre un ulteriore sviluppo è rappresentare graficamente e contemporaneamente sia le tracce inerenti ai predicted boxes che a quelle delle masks associate allo stesso oggetto, in modo da verificare il corretto tracciamento di entrambe nel flusso ottico.