



POLITECNICO DI TORINO

COMPUTER VISION
04ISFOV

TESINA

Rilevazione superamento di corsia

Autori:

Fabiola Polidoro

Davide Trenti

Matricola:

231594

224714

Docente:

prof. Luigi De Russis

A.A. 2016/2017

1 Introduzione

Il *lane departure warning system* è un meccanismo che avvisa il guidatore quando il suo veicolo sta uscendo dalla corsia di marcia, in modo da minimizzare gli incidenti dovuti ad errori, distrazioni o sonnolenza. Il tipo più semplice di *lane departure warning system* è quello che si limita ad avvertire il guidatore, in caso di superamento di corsia, mediante segnali visivi, uditivi oppure vibrazioni.

Il rilevamento delle strisce stradali avviene mediante una telecamera posta sul cruscotto della vettura e sensori laser e infrarossi montati davanti oppure sotto al veicolo.

1.1 Obiettivo e requisiti della tesina

L'obiettivo della tesina è quello di sviluppare un'applicazione desktop in grado di analizzare video ripresi da un'auto in movimento e di rilevare le due linee laterali della corsia, segnalando gli eventuali *lane crossing*. In particolare l'applicazione deve essere in grado di operare correttamente su video ripresi in condizioni di bel tempo sia di giorno che all'imbrunire oppure di notte su strade dotate di illuminazione.

I capitoli seguenti di questa relazione descriveranno i passi dello sviluppo e gli algoritmi utilizzati, in modo che questa tesina possa anche fungere come una sorta di documentazione per il programma stesso.

1.2 Strumenti utilizzati

Per lo sviluppo è stato utilizzato JavaFX che, oltre al Java, aggiunge JavaFX script e librerie grafiche per poter sviluppare interfacce utente.

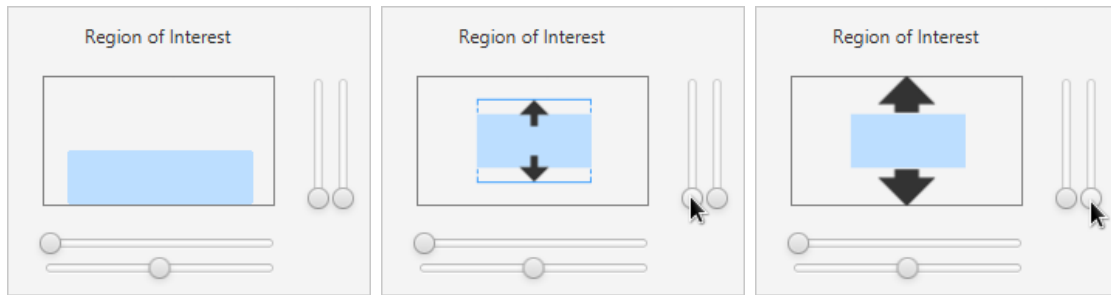
Per gli algoritmi di computer vision invece è stata utilizzata la libreria OpenCV.

2 Implementazione software

2.1 Interfaccia grafica

Il programma è dotato di un'interfaccia grafica composta principalmente da una barra dei menu, un'area in cui viene mostrato il video e un pannello sottostante contenente i controlli (*stop*, *play*, *rewind*, *fast forward*, *slow*, *next frame*, *previous frame*) e informazioni come il nome del file, il numero di frame corrente e l'elapsed time.

Cliccando sulla voce “*Settings*” nella barra del menu è possibile rendere visibile un pannello che permette di modificare quasi tutti i parametri relativi al riconoscimento delle linee di corsia e alla dimensione e posizionamento della *region of interest* (ROI) su cui gli algoritmi lavorano. Nella parte alta di tale pannello è presente un riquadro, che si adatta automaticamente all'aspect ratio del file video aperto, contornato da quattro slider che permettono di modificare la dimensione e la posizione della ROI, rappresentata come rettangolo azzurrino. Passando il mouse su ciascuno di questi slider, il contenuto del riquadro cambia temporaneamente per mostrare l'azione legata al comando.



(a) Posizione e dimensione corrente della ROI. (b) Aumenta/diminuisci l'altezza della ROI. (c) Sposta la ROI più in alto/basso.

Figura 1: Esempio di suggerimenti visivi relativi al posizionamento e alla dimensione verticali della *region of interest*.

Degno di nota è anche l'elemento `TitledPane` “Display”, che contiene una serie di checkbox che permettono di abilitare e disabilitare la visualizzazione sul video dei vari passi compiuti dall'algoritmo di *lane detection*, come ad esempio i contorni, le strisce rilevate, quelle selezionate dall'algoritmo come facenti parte di una corsia e via dicendo.

3 Lane detection

Questo capitolo descrive come viene rilevata la corsia, supponendo che l'utente abbia già provveduto a selezionare un file video mediante il menu **File**→**Open** e ad eventualmente riposizionare la *region of interest*.

È opportuno sottolineare che tutte le operazioni descritte nei paragrafi successivi vengono eseguite unicamente all'interno della ROI, pertanto occorre che essa sia posizionata in modo che il suo centro corrisponda al centro della corsia in cui si trova il veicolo e la parte superiore corrisponda all'orizzonte.

3.1 Edge detection

La prima azione eseguita su ciascun frame è la *edge detection*, cioè il rilevamento dei bordi dell'immagine. Per questa operazione il frame deve essere convertito in scala di grigi mediante il metodo di OpenCV `cvtColor` con parametro `CV_BGR2GRAY` e viene applicata una riduzione del rumore con il metodo `blur`.

In seguito viene applicato il metodo `Canny`, che dall'immagine in scala di grigi ne genera una binaria in cui il bianco rappresenta i bordi e il nero rappresenta tutto ciò che bordo non è.

La scelta del `Canny` rispetto ad altri metodi si è rivelata più efficace durante le prove.

Sia i parametri del `Canny` che quelli del `blur` hanno un valore di default preimpostato, modificabile manualmente tramite gli appositi slider nella sezione “*Canny*” del pannello “*Settings*”.

3.2 Line detection

Una volta ottenuti i contorni dell'immagine si procede con l'individuazione delle linee mediante la trasformata di Hough probabilistica (`HoughLinesP`). Quest'ultima è stata scelta in quanto computazionalmente più efficiente rispetto alla trasformata di Hough semplice¹, dato che anziché considerare tutti i punti di bordo dell'immagine, ne considera solo un sottoinsieme permettendo di ridurre il tempo di esecuzione. I parametri del metodo `HoughLinesP` sono impostati ad un valore di default trovato

¹La trasformata di Hough semplice opera su ciascun punto dell'immagine, trovando la famiglia di linee passanti per esso, che dà luogo ad una sinusoide. Due punti distinti giacciono sulla stessa linea se le curve che li attraversano intersecano lo stesso piano. I parametri della linea individuata sono dati dalle coordinate polari ρ e θ del punto in cui le curve si incontrano. Per una spiegazione più dettagliata consultare la [documentazione di OpenCV](#).

empiricamente, ma sono anche modificabili manualmente dall'utente attraverso gli slider della sezione "*Hough properties*" del pannello "*Settings*".

3.3 Scelta delle linee e disegno della corsia

La trasformata di Hough probabilistica restituisce un array di linee, individuate dalle coordinate delle due estremità (x_1, y_1) e (x_2, y_2) ma, ai fini della *lane detection*, non tutte sono utili.

Le linee vengono scelte in base ad alcuni parametri che possono essere calibrati tramite gli slider della sezione "*Lane constraints*" del pannello "*Settings*":

- pendenza ("*slope*") minima e massima. Una linea è considerata come probabile striscia stradale se la sua pendenza tende ad essere perpendicolare rispetto all'orizzonte ed è compresa in un intervallo impostato per default tra 10 e 90°. In questo modo vengono escluse le linee orizzontali o quasi orizzontali.
- numero di linee candidate ("*number of candidates*"). Per ciascuna linea, viene considerato l'estremo inferiore e vengono valutate la posizione e la distanza di quest'ultimo dal centro della *region of interest*. In base alla posizione la linea è classificata come sinistra o destra e, in base alla distanza dell'estremo dal centro, è inserita in un `arrayList` ordinato, `leftList` oppure `rightList`, dopo essere stata estesa fino ai bordi della ROI. Entrambi gli `arrayList` vengono potati, mantenendo solo le prime n linee ("*candidates*"), i cui estremi inferiori sono più vicini al centro.
- angolo ("*alpha*") formato da una coppia di linee. Le linee presenti nei due `arrayList` `leftList` e `rightList` vengono combinate tra loro per formare coppie `<linea_sinistra, linea_destra>`. Tra esse vengono mantenute solo le coppie il cui angolo è compreso tra l'intervallo specificato dagli slider "*Min alpha*" e "*Max alpha*", che per default sono impostati rispettivamente a 65° e 120°.
- distanza del punto di fuga delle linee dal centro della ROI ("*Horizon radius*"). Una seconda potatura della lista di coppie candidate come corsia viene fatta considerando la distanza tra il punto di intersezione di ciascuna coppia e l'estremo superiore del centro della *region of interest*. Questo vincolo si è reso necessario per evitare che la presenza di frecce o altre scritte (ad esempio la lettera "A") sul manto stradale potesse trarre in inganno il programma e produrre un fastidioso *flickering* della corsia evidenziata. Pertanto, solo le coppie la cui intersezione si trova entro un certo raggio dal-

l'estremo superiore del centro della ROI vengono accettate come potenziale corsia.



Figura 2: Esempio di simboli e scritte che causano il *flickering* della corsia. Le linee azzurre rappresentano le coppie candidate.

Tra tutte le coppie, sarà scelta come candidata più probabile quella il cui *alpha* si discosta meno da quello della coppia scelta nel frame precedente. Ciò permette di ridurre il *flickering* della corsia quando il veicolo si trova su strade a più corsie.



Figura 3: Esempio di flickering della corsia dovuto ad *alpha* troppo diversi tra un frame e l'altro.

Prima di essere disegnata a video, la corsia deve rimanere stabile per un certo numero di frame, impostati dall'utente mediante lo slider "*frames before showing lane*".

Se nessuna coppia di linee rispetta i vincoli imposti, viene disegnata l'ultima corsia valida rilevata nei frame precedenti. Il numero di tali frame è anch'esso impostabile tramite l'apposito slider "*frames before hiding lane*". Se l'ultima corsia valida si trova nel passato, oltre il numero di frame specificati, nel frame corrente non viene più disegnata, in quanto si presuppone che il veicolo si trovi su un tratto di strada privo di segnaletica.

3.4 Lane crossing

Una volta individuate le linee che delimitano la corsia, viene calcolata la distanza tra il centro della *region of interest* e l'estremità inferiore di ciascuna delle due strisce. Tali distanze, chiamate `leftDelta` e `rightDelta` vengono confrontate tra loro e, quella con valore minore, viene utilizzata per decidere il colore da assegnare al riempimento della corsia. Il colore è definito come oggetto di tipo `Scalar`, il cui

costruttore accetta tre parametri interi corrispondenti ai valori BGR, che in questo caso saranno impostati rispettivamente a:

- (l) (0, leftDelta, 255 - leftDelta) per la striscia sinistra;
- (r) (0, rightDelta, 255 - rightDelta) per quella destra.

Il lane crossing vero e proprio viene mostrato all'utente tramite il colore con cui viene riempita l'area compresa tra le due strisce, che può corrispondere, come già accennato, al caso (l) o (r), a seconda di chi tra `leftDelta` e `rightDelta` è minore. Quando il veicolo si trova esattamente all'interno della corsia il colore è verde e, man mano che si avvicina alla linea di mezzzeria, il colore sfuma via via verso l'arancione fino a raggiungere il rosso quando il veicolo si trova a cavallo della striscia.



Figura 4: *Lane crossing* verso sinistra evidenziato con i colori.

3.5 Risultati

L'algoritmo è stato testato principalmente con video girati con la nostra *dash cam* su strade urbane, extraurbane e tangenziali, nelle condizioni richieste dai requisiti della tesina, pertanto i parametri di default sono stati impostati per garantire il miglior funzionamento con essi.

In aggiunta, sono stati usati anche filmati forniti dal docente per verificare l'adattività dei parametri a video girati con dispositivi diversi dal nostro.

Di seguito sono riportati i commenti su ciascun file.

- `2756_giorno.avi`. La *lane detection* funziona correttamente per quasi tutta la durata del video: per alcuni istanti la corsia non viene rilevata quando il veicolo si trova in prossimità dell'uscita dalla galleria urbana e durante il lane crossing in prossimità del semaforo. Ciò è molto probabilmente dovuto al repentino cambio di illuminazione dell'intera scena, per il primo caso, e alla eccessiva vicinanza del nostro veicolo all'auto ferma al rosso.
- `2873_giorno.avi`. Anche in questo caso l'algoritmo non dà particolari problemi per la maggior parte del tempo. In alcuni tratti di curva, la corsia non

viene rilevata a causa di angoli “*alpha*” troppo grandi rispetto a quelli impostati di default. Aumentando il parametro “*max alpha*”, si verifica un fastidioso “*flickering* di corsia” appena il veicolo giunge su strada a tre corsie, pertanto si è preferito avere questi vuoti in curva piuttosto che rilevamenti sbagliati sui rettilinei.

Oltre agli sfarfallii dovuti alla segnaletica stradale, si ottengono effetti simili anche quando il veicolo passa accanto a TIR o bus, che proiettano un’ombra lunga sulla carreggiata.

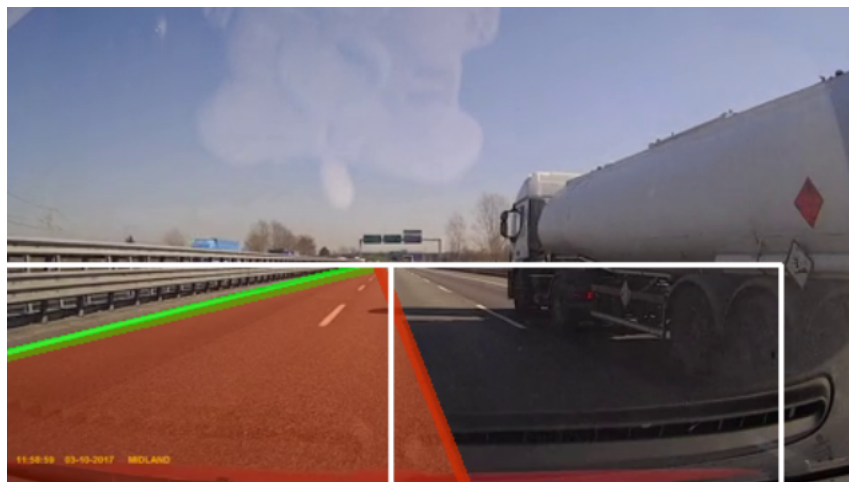


Figura 5: Rilevamento errato della corsia dovuto all’ombra di un TIR.

- **2980_notte.avi.** Questo filmato ripercorre, di notte, lo stesso tratto di strada del filmato 2873_giorno.avi e, fino all’immissione nella tangenziale vera e propria, la corsia viene sempre rilevata in modo corretto, a parte gli sfarfallii dovuti alla segnaletica stradale già descritti in precedenza.
L’algoritmo incontra difficoltà nel rilevare sempre la corsia quando l’illuminazione stradale è assente (tratti extraurbani), anche se nel complesso le prestazioni si possono considerare accettabili.
- **2796_imbrunire.avi e 2797_imbrunire.avi.** Per entrambi i filmati l’algoritmo funziona perfettamente per tutta la durata, nonostante nel secondo video l’illuminazione stradale sia abbastanza scarsa. Occasionalmente si verificano piccoli sfarfallii dovuti a segni di frenata sull’asfalto che però non invalidano il rilevamento della corsia.
- **V4_highway.mp4, V3_cross1.mp4 e V3_cross2.mp4.** I parametri di default si adattano bene ai tre video, nonostante siano comunque presenti sfarfallii di corsia dovuti a segni di frenata sull’asfalto, molto più evidenti rispetto al video 2796_imbrunire.avi poichè l’asfalto ha un colore molto più chiaro.

- **V2_highway.mp4.** La *lane detection* in questo filmato dà pessimi risultati con i parametri di default, in quanto la corsia non viene mai rilevata. Impostando il parametro “*Max alpha*” a 150 si ottiene un risultato accettabile anche se con frequenti sfarfallii di corsia.
- **V1_regular.mp4.** Anche con questo video i parametri di default non vanno bene, poichè la corsia non viene rilevata in prossimità delle curve. Queste ultime sono molto strette e causano un eccessivo spostamento del punto di incontro delle linee destra e sinistra dal punto centrale superiore della *region of interest*.

In questo caso, per ottenere un rilevamento corretto, oltre che impostare il parametro “*Max alpha*” a 150, occorre anche modificare il parametro “*Horizon variance*” impostandolo a 90.

4 Migliorie

Nella sua versione attuale il programma può essere adattato manualmente per l'utilizzo con filmati molto diversi tra loro in termini di qualità video ed illuminazione, tuttavia sarebbe possibile migliorarlo ulteriormente implementando le seguenti funzionalità:

- adattività automatica dei parametri (ad esempio Canny, Blur, trasformata di Hough) in base alla risoluzione del video, alla qualità e all'illuminazione;
- riconoscimento automatico del punto di fuga e ridimensionamento automatico della *region of interest* in base alla larghezza della corsia, così che l'algoritmo di *lane detection* si comporti correttamente con qualsiasi tipo di curva e in presenza di salite e discese ripide che fanno variare l'altezza dell'orizzonte;
- gestione e rimozione di crepe e segni di frenata sull'asfalto, tramite filtraggio dei colori prima dell'applicazione del Canny;
- evidenziazione della corsia con un contorno preciso e non triangolare, mediante suddivisione della *region of interest* in sub-ROI, o, in alternativa, riempimento con gradiente di trasparenza verticale;
- estensione del supporto a filmati ad alta risoluzione: l'applicazione attuale non è in grado di riprodurre tali video ad un frame rate accettabile, infatti si è reso necessario ridurre di quattro volte i filmati full HD ripresi con la nostra *dash cam*.

Risultati molto migliori possono essere ottenuti con tecniche che utilizzano cambi di prospettiva (*Bird's eye view*) per rendere parallele tra loro le linee di corsia e permetterne una manipolazione più accurata e *machine learning* per riconoscere la segnaletica sull'asfalto e ridurre gli errori nel riconoscimento delle corsie.