

UNIVERSITÀ DEGLI STUDI DI PARMA
FACOLTÀ DI INGEGNERIA

CORSO DI LAUREA IN INGEGNERIA INFORMATICA

CONTROLLO AUTOMATICO DI FARI DI VEICOLI
MEDIANTE VISIONE ARTIFICIALE

Relatore:

Chiar.mo Prof. Alberto Broggi

Correlatori:

Dr. Nereo Pallaro

Ing. Pietro Cerri

Candidato:

Luca Bombini

ANNO ACCADEMICO 2003-2004

*Alla mia famiglia
e ad Annalisa*

Indice

Introduzione	1
1 Analisi del problema	4
1.1 Il veicolo utilizzato	4
1.2 Descrizione del problema	5
1.2.1 Caratteristiche degli scenari	6
1.2.2 Caratteristiche generali dei fasci luminosi dei fari	8
1.2.3 Uso del filtro rosso	9
1.3 Tecniche possibili per la risoluzione del problema	10
2 Preelaborazione delle immagini in ingresso	13
2.1 Metodo della doppia Binarizzazione	14
2.2 Metodo della Differenza	15
2.3 Unione dei metodi di Binarizzazione e Differenza	19
3 Individuazione dell'andamento della strada	22
3.1 Uso della trasformata di Hough	22
3.1.1 Ricerca di rette	23
3.1.2 Illuminazione stradale	25
3.1.3 Linee della carreggiata	29
3.2 Considerazioni sull'applicabilità	31
3.2.1 Prestazioni del rilevamento della posizione della carreggiata .	33

4	Caratterizzazione dei fari	36
4.1	Modellazione dei fari	36
4.1.1	Modelli per immagine binarizzata con soglia alta	37
4.1.2	Modelli per immagine differenza	37
4.2	Algoritmo di Pattern Matching	38
4.3	Prestazioni dell'individuazione di un faro	41
5	Tracking dei fari	43
5.1	Algoritmo di Tracking	44
5.1.1	Predittore	44
5.1.2	Etichettatore	46
5.2	Valutazione delle prestazioni dell'algoritmo di tracking	47
5.3	Determinazione della presenza di un veicolo	49
6	Risultati sperimentali	51
6.1	Tempi di esecuzione	51
6.2	Prestazioni quantitative	53
6.3	Prestazioni sulla distanza di localizzazione	59
7	Conclusioni	62
7.1	I limiti di questa analisi	63
7.2	Sviluppi futuri	64
7.2.1	Condizioni ambientali differenti	64
7.2.2	Restringimento dell'inquadratura	64
7.2.3	Validazione delle prestazioni	66
7.2.4	Flusso ottico	67
A	Specifiche della telecamera utilizzata	69
A.1	Descrizione generale	69
A.2	Descrizione tecnica	69
B	Manuale di utilizzo del software	71

Indice iv

Bibliografia **78**

Ringraziamneti **80**

Elenco delle figure

1.1	Veicolo sperimentale.	5
1.2	Camera NC5000.	5
1.3	Segnaletica orizzontale.	6
1.4	Cartelli posizionati a terra sul lato destro della carreggiata.	6
1.5	Lampioni.	7
1.6	Guard rail a lato della carreggiata.	7
1.7	Riflessione dei fari anabbaglianti.	8
1.8	Riflessione dei fari abbaglianti.	8
1.9	Acquisizione con abbaglianti spenti.	8
1.10	Acquisizione con abbaglianti accesi.	8
1.11	Acquisizione senza filtro rosso.	9
1.12	Acquisizione con filtro rosso.	9
2.1	Regione dell'immagine da preelaborare.	14
2.2	Istogramma (colore rosso) della Fig. 2.1.	16
2.3	Istogramma delle intensità (colore verde) della Figura 2.1.	16
2.4	Confronto fra istogrammi dopo 3 iterazioni del filtro di media.	17
2.5	Confronto fra istogrammi dopo 6 iterazioni del filtro di media.	17
2.6	Immagine differenza.	18
2.7	Estrapolazione dell'immagine differenza.	19
2.8	Immagine binarizzata con soglia alta.	21
2.9	Immagine differenza binarizzata.	21
3.1	Linea parametrica.	24

3.2	Tre punti allineati.	25
3.3	Trasformata di Hough.	25
3.4	Porzione d'immagine dove viene applicata la trasformata di Hough.	25
3.5	Contorni dell'illuminazione stradale.	26
3.6	Spazio accumulatore.	26
3.7	Rappresentazione delle linee individuate.	27
3.8	Limitazione della regione d'interesse.	28
3.9	Porzione d'immagine dove viene applicata la trasformata di Hough.	29
3.10	Contorni delle linee della carreggiata.	29
3.11	Contorni delle linee della carreggiata dopo l'erosione.	30
3.12	Spazio accumulatore.	30
3.13	Rappresentazione delle linee individuate.	30
3.14	Limitazione della regione d'interesse.	31
3.15	Porzione d'immagine dove viene applicata la trasformata di Hough.	32
3.16	Spazio Accumulatore.	32
3.17	Contorni delle linee della carreggiata.	32
4.1	Modelli dei fari.	37
4.2	Modelli dei fari.	38
4.3	Immagine di test.	39
4.4	(a) Immagine binarizzata con soglia alta, (b) Immagine differenza.	39
4.5	(a) Immagine binarizzata con soglia alta, (b) Immagine differenza.	40
4.6	Immagine di test.	40
4.7	Risultati dell'elaborazione dell'immagine	40
5.1	Diagramma di flusso dell'algoritmo di <i>tracking</i>	44
5.2	Immagine differenza con localizzazione dei fari.	45
5.3	Zona calcolata dal predittore.	45
5.4	Etichettatura dei fari localizzati.	46
5.5	Entrambi i fari di un veicolo hanno la stessa etichetta.	47
5.6	Immagine acquisita dalla camera.	48
5.7	Errore dell'etichettatore.	48

6.1	Icona fari abbaglianti accesi.	54
6.2	Icona fari abbaglianti accesi.	54
6.3	Esempio di falso positivo.	54
6.4	Caso tipico di corretta rilevazione di una serie di veicoli.	55
6.5	Risultato corretto dell'algoritmo.	55
6.6	Riconoscimento di un veicolo a notevole distanza.	55
6.7	Veicolo con faro posteriore non funzionante	55
6.8	Corretto controllo dei fari in caso di sorpasso.	56
6.9	Corretto riconoscimento di un motociclo.	56
6.10	Corretto controllo dei fari in caso di sorpasso.	56
6.11	Riconoscimento di un veicolo a lato in curva.	56
6.12	Riconoscimento percorrendo un cavalcavia	57
6.13	Riconoscimento percorrendo una curva	57
6.14	Caso di falso positivo: non è presente nessun veicolo.	58
6.15	Falso positivo.	58
6.16	Corretta rilevazione di un veicolo a distanza considerevole.	58
6.17	Controllo corretto dei fari.	58
6.18	Riconoscimento di diversi veicoli contemporaneamente.	59
6.19	Corretto riconoscimento in situazione di sorpasso.	59
6.20	Riconoscimento di un veicolo notevolmente distante.	59
6.21	Corretto riconoscimento di un veicolo in curva.	59
7.1	Limiti del programma	63
7.2	In rosso viene indicato la distanza del veicolo incrociato.	66
7.3	Il radar funziona efficacemente anche con più veicoli inquadrati.	66
7.4	Rappresentazione del flusso ottico ricavato dal movimento della sfera.	67
7.5	Rappresentazione dello spostamento di un oggetto.	67
A.1	NC5000 CLinkCam	69
B.1	Interfaccia grafica di Gold.	72
B.2	Pannello principale.	72

B.3	Pannello preelaborazione	74
B.4	Pannello localizzazione dei fari.	75
B.5	Pannello di individuazione della posizione della carreggiata.	76

Elenco delle tabelle

3.1	Prestazioni sull'individuazione della carreggiata con abbaglianti spenti.	33
3.2	Prestazioni con abbaglianti spenti.	33
3.3	Prestazioni sull'individuazione della carreggiata con abbaglianti accesi.	34
3.4	Prestazioni con abbaglianti accesi.	34
4.1	Prestazioni sulla distanza di individuazione con abbaglianti spenti. . .	41
4.2	Prestazioni sulla distanza di individuazione con abbaglianti accesi. . .	41
6.1	Caratteristiche tecniche del calcolatore utilizzato per i test.	51
6.2	Sequenze analizzate durante i test.	52
6.3	Tempi d'elaborazione di un <i>frame</i> per ciascuna sequenza.	52
6.4	Tempi d'elaborazione di un <i>frame</i> per ciascuna sequenza.	53
6.5	Errori nell'elaborazione con fari abbaglianti spenti.	54
6.6	Errori nell'elaborazione con fari abbaglianti spenti.	57
6.7	Distanza di riconoscimento di un veicolo con fari abbaglianti spenti. .	60
6.8	Distanza di riconoscimento di un veicolo con fari abbaglianti accesi. .	60
7.1	Tempi d'elaborazione di un <i>frame</i> per ciascuna sequenza.	65
7.2	Tempi d'elaborazione di un <i>frame</i> per ciascuna sequenza.	65

Introduzione

Il controllo intelligente dei fari di veicoli è un argomento di ricerca vasto e complesso nel quale prestigiose case automobilistiche di tutto il mondo stanno impegnando ingenti risorse. Tale impegno non sorprende. Secondo i dati raccolti dall'Ufficio Federale Tedesco di Statistica, più del 40% di tutti gli incidenti mortali avviene di notte, nonostante in quella ora circoli fino all'80% di veicoli in meno rispetto alle ore diurne. Inoltre studi scientifici hanno rilevato che la percezione visiva umana, con la quale assorbiamo il 90% delle principali informazioni relative al traffico, si riduce del 4% in condizioni di scarsa visibilità come quelle notturne. Per queste ed altre ragioni una efficiente illuminazione del percorso stradale è determinante nel garantire maggiore sicurezza in condizioni in cui la visibilità non è ottimale.

L'obiettivo di questa tesi è controllare in maniera completamente *automatica* ed *intelligente* l'accensione e lo spegnimento dei fari abbaglianti di un generico veicolo. Per *automatico* intendiamo che, una volta abilitato, il controllore dei fari si occupi di accendere e spegnere i fari abbaglianti senza nessun intervento da parte dell'utente. Per *intelligente* si ci riferisce al corretto uso dei fari, garantendo la massima illuminazione della carreggiata ed evitando l'abbagliamento di un veicolo vicino. Esistono diversi percorsi stradali, come curve strette e avvallamenti con pendenze elevate, in cui il controllo manuale dei fari abbaglianti risulta molto scomodo e pericolosamente lento. Percorrendo una curva, potrebbe essere complicato e irrimediabilmente lento, incrociando un veicolo, spegnere gli abbaglianti avendo entrambe le mani impegnate nella corretta impostazione della curva. Un'altra situazione di pericolo si verifica quando il guidatore dimentica i fari abbaglianti accesi all'incrocio con un altro veicolo. Il software prodotto mira a scongiurare queste e altre situazioni critiche per la

sicurezza in ambienti notturni.

In questo particolare ambito di ricerca è nata una collaborazione tra la Divisione Tecnologie Ottica e Fotonica del Centro Ricerche Fiat di Trento e il Dipartimento di Ingegneria dell'Informazione dell'Università di Parma. Il C.R.F. ha messo a disposizione sequenze di immagini notturne inquadranti diversi percorsi stradali. Tali filmati che sono stati acquisiti appositamente mediante una telecamera installata dietro il parabrezza di una automobile sperimentale, hanno costituito la base sulla quale è stato sviluppato tutto l'algoritmo per il controllo dei fari anteriori del veicolo.

Scopo ultimo di questo progetto, denominato Incrocio Veicoli, è di pilotare direttamente lo stato degli abbaglianti.

Il programma è stato scritto in linguaggio C e sviluppato all'interno di GOLD¹ [4], una piattaforma per lo sviluppo di applicazioni per la visione, realizzato presso il Laboratorio di Visione Artificiale dell'Università degli Studi di Parma.

Nel dettaglio, il contenuto di questa tesi è il seguente:

- Nel Capitolo 1 vi è un'analisi approfondita del problema: verranno illustrate tutte le difficoltà relative ai diversi percorsi stradali, alle diverse tipologie di fari e alle svariate tecniche per il riconoscimento mediante la visione artificiale di veicoli in prossimità.
- Dal Capitolo 2 inizia la descrizione vera e propria del lavoro svolto. Verranno illustrati diversi metodi di preelaborazione applicati alle immagini in ingresso. Scopo principale della preelaborazione è cercare di mettere in evidenza i fari dei veicoli e scartare illuminazione stradale e segnalatori di carreggiata. Questa fase del progetto ha richiesto un notevole impegno perchè è fondamentale per il corretto funzionamento dell'algoritmo.
- Nel Capitolo 3 si prende in esame la necessità di riconoscere l'andamento della strada. Si analizzerà una particolare trasformata matematica, detta trasformata di Hough, che permette l'individuazione di linee nell'immagine. Tale operazione

¹*Generic Obstacle and Lane Detection*

è necessaria per aumentare le prestazioni sulla distanza di localizzazione e sull'affidabilità della rilevazione di un veicolo.

- Il Capitolo 4 prende in esame il problema del riconoscimento di fari nell'immagine. L'algoritmo utilizzato è quello del *pattern matching*. Tale algoritmo consente di individuare fari di forma e intensità diverse fra loro.
- Il Capitolo 5 illustra un particolare algoritmo che permette di inseguire i fari durante il filmato e studiarne il movimento, controllando per quanto tempo un determinato faro è presente nella scena. Questo approccio permette di discriminare fra l'illuminazione generata dai fari e quella riflessa o proiettata dagli illuminatori stradali.
- Il Capitolo 6 è dedicato all'esposizione dei risultati ottenuti. Per prima cosa verranno mostrati i risultati quantitativi di una serie di prove. Successivamente verranno esposte le prestazioni in termini di tempo impiegato dall'algoritmo ad individuare un veicolo in un filmato e la distanza corrispondente all'istante di rilevazione. Inoltre si prenderanno in esame i risultati ottenuti nelle situazioni più critiche per l'algoritmo.
- Con il Capitolo 7 si conclude l'analisi dell'algoritmo, illustrandone i pregi, i limiti ed i possibili sviluppi futuri.
- L'Appendice A descrive nel dettaglio le caratteristiche tecniche della camera utilizzata nella sperimentazione.
- L'Appendice B contiene le istruzioni per utilizzare il modulo *software* realizzato per questo lavoro di tesi.

Capitolo 1

Analisi del problema

Questo capitolo contiene una approfondita analisi del problema affrontato in questa tesi, con una breve descrizione del veicolo utilizzato per l'acquisizione. Di particolare importanza è la parte relativa allo studio degli scenari possibili e alle tecniche sperimentate e valutate per una corretta individuazione del veicolo.

1.1 Il veicolo utilizzato

Il veicolo utilizzato per le riprese è una Lancia Kappa dotata di proiettori anteriori adattativi; si tratta di fari di nuova concezione, aventi la caratteristica di illuminare la strada secondo la direzione determinata dal movimento dello sterzo. Il fatto che questo mezzo monti tale sistema di illuminazione non si è dimostrato uno svantaggio per il nostro algoritmo.

La camera utilizzata per l'acquisizione delle immagini è una NC5000 prodotta dalla Neuricam. Posizionata mediante un supporto regolabile, alloggia all'interno del veicolo tra il parabrezza e lo specchietto retrovisore. Accenniamo solo al fatto che questa camera è di tipo CMOS ed ha una risoluzione di 640x480 pixel (VGA) a toni di grigio con risposta logaritmica. L'Appendice A prende in esame nel dettaglio le specifiche tecniche della camera.

La camera è collegata direttamente ad un calcolatore tramite una scheda di acquisizione PCI. Tramite il *software* fornito con la camera è possibile regolare diversi



Figura 1.1: Veicolo sperimentale.



Figura 1.2: Camera NC5000.

parametri della finestra di acquisizione: dimensione, *frame rate* e risoluzione geometrica. In fase di sperimentazione le immagini acquisite vengono memorizzate sequenzialmente e date in ingresso all'algoritmo di controllo abbaglianti che produce in tempo reale il comando di accensione o spegnimento dei fari abbaglianti. Fino a questo momento, però, non è ancora stato possibile testare l'algoritmo nella modalità descritta sopra. Il lavoro si è fermato con l'implementazione software del controllore dopo aver processato un notevole numero di filmati precedentemente acquisiti. Nonostante ciò, dato che l'algoritmo è destinato ad una sperimentazione sul veicolo, sono state considerate tutte quelle impostazioni che intervengono nella sperimentazione in tempo reale, quali ad esempio, il tempo che trascorre fra l'acquisizione di due distinte immagini.

1.2 Descrizione del problema

Il problema affrontato in questa tesi è il progetto e l'implementazione di un algoritmo capace di individuare, se presenti, veicoli in ambiente stradale in condizioni notturne. Il problema non è banale se si considera che in un generico percorso stradale strutturato, oltre ai fari di veicoli, sono presenti lampioni di ogni genere, segnalatori intermittenti di pericolo, cartelli e paletti catarifrangenti. L'algoritmo, deve essere in grado sia di rilevare correttamente un veicolo sia di identificare una situazione di assenza di veicoli con un numero di errori molto basso.

É fondamentale specificare cosa si intende per veicolo in prossimità onde evitare di abbagliare pericolosamente un altro guidatore. Il C.R.F. non ha fornito precise specifiche riguardo alla distanza di localizzazione. E' tuttavia necessario che tali distanze siano, massimizzate trovando un compromesso fra velocità di elaborazione e prestazioni in termini di errori di riconoscimento. Queste semplici considerazioni, inoltre, variano a seconda delle condizioni atmosferiche presenti. I successivi capitoli presenteranno in maniera esauriente, quali sono stati i metodi utilizzati per la risoluzione del problema.

1.2.1 Caratteristiche degli scenari

In questo paragrafo illustriamo le principali caratteristiche delle immagini acquisite. Vengono acquisiti quattro *frame* per secondo. Ogni *frame* ha dimensione 640x478 pixel, con risoluzione radiometrica di 10 bit. Le sequenze, sono state acquisite esclusivamente di notte. Anche se la sperimentazione lo prevede a breve termine, fino ad ora non sono stati processati filmati dove le condizioni ambientali presentassero nebbia, pioggia o una leggera luce solare.

Riportiamo ora alcune immagini acquisite con il veicolo in movimento inquadranti diversi percorsi stradali. Queste immagini sono state convertite a otto bit esclusivamente per permetterne la visualizzazione ed averne così una visione qualitativa. Tali immagini, così convertite non saranno utilizzate dall'algoritmo: i paragrafi successivi spiegheranno le tecniche di conversione adottate.



Figura 1.3: Segnaletica orizzontale.



Figura 1.4: Cartelli posizionati a terra sul lato destro della carreggiata.



Figura 1.5: Lampioni.



Figura 1.6: Guard rail a lato della carreggiata.

Le immagini in Figura 1.3 ed 1.5 sono rappresentative di quanto la riflessione della luce dei catarifrangenti e dell'illuminazione stradale sia evidente nelle immagini e sia paragonabile a quella dei fari delle vetture sia per forma che per dimensione.

Le immagini in Figura 1.4 ed 1.6 illustrano come la segnaletica orizzontale ed i guard rail possano riflettere la luce in maniera inaspettata.

Tutti questi elementi portano alla conclusione che per distinguere un fanale da altra illuminazione non sarà sufficiente considerarne solo l'intensità, la forma, la dimensione o la posizione nell'immagine. Paletti catarifrangenti corrispondono a fari valutando solo le caratteristiche sopra. Sarà quindi necessario riuscire a capire come sia possibile preelaborare le immagini al fine di evidenziare i fari di veicoli.

Un'altra importante considerazione da fare è che l'immagine che acquisisce la telecamera varia sensibilmente a seconda che gli abbaglianti del veicolo siano accesi oppure no. Infatti l'intensità luminosa dei fasci di luce riflessi dalla segnaletica stradale è direttamente proporzionale all'intensità del raggio incidente. Le Figure 1.7 ed 1.8 riportano due immagini che illustrano in maniera qualitativa la differenza nei due casi.

Come si nota in Figura 1.8, l'intensità della riflessione dei paletti a lato della carreggiata, sebbene siano a distanza considerevole, è superiore che nel caso con abbaglianti spenti. Questo aumento sarà tanto maggiore quanto i paletti sono vicini al veicolo entro il raggio di illuminazione dei fari.

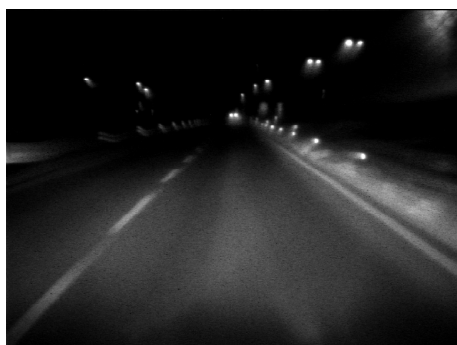


Figura 1.7: Riflessione dei fari anabbaglianti.

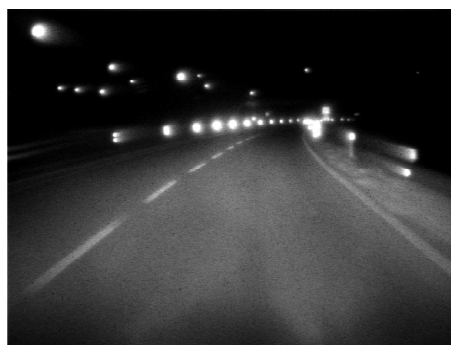


Figura 1.8: Riflessione dei fari abbaglianti.

1.2.2 Caratteristiche generali dei fasci luminosi dei fari

Una analisi delle intensità luminose dei fari è fondamentale per capire se sia possibile in qualche modo distinguerli da quelli della segnaletica stradale. Una considerazione importante da fare è quella che si nota nelle immagini in Figura 1.9 e 1.10. Queste figure sono state acquisite staticamente, inquadrando veicoli a distanze note. In particolare il veicolo nella corsia di sinistra è distante 50 metri, mentre quello di destra, è distante 150 metri.



Figura 1.9: Acquisizione con abbaglianti spenti.



Figura 1.10: Acquisizione con abbaglianti accesi.

Anche se qualitativamente è difficile notarlo la riflessione dei paletti nella Figura 1.10 è superiore all'intensità dei fari posteriori del veicolo a distanza 150 metri. Questo effetto aiuta a capire che non è possibile distinguere fari posteriori a tale distanza solo basandosi sulle informazioni di intensità e forma del fascio. Sarà necessario introdurre

qualche metodo che elimini gli oggetti riflettenti a bordo della carreggiata andando ad analizzare le linee della carreggiata e il movimento del faro in una serie di immagini successive. Questi approcci risulteranno determinanti nelle prestazioni dell' algoritmo sulla distanza e sulla precisione della localizzazione.

Un'altra considerazione importante da fare riguarda la differenza dell'intensità luminosa percepita dalla camera nel caso inquadri un veicolo frontalmente o posteriormente. Infatti i fari anteriori sono molto più luminosi di quelli posteriori. Questa sensibile differenza renderà necessario diversificare i parametri che controllano la parte di algoritmo che si occupa della localizzazione dei fari nell'immagine; la luminosità dei fari posteriori varia anche a seconda che il veicolo stia frenando¹ o meno.

1.2.3 Uso del filtro rosso

Per le motivazioni indicate nel paragrafo precedente si è studiato un metodo che permettesse di esaltare la luminosità dei fari posteriori delle vetture rispetto a quella della segnaletica stradale. Essendo questi obbligatoriamente di colore rosso, si è montato sull'obbiettivo della camera un filtro rosso [8]. Tale sistema, ha dato buoni risultati.

Riportiamo ora alcune immagini acquisite staticamente inquadranti veicoli a distanza predefinita.



Figura 1.11: Acquisizione senza filtro rosso. Figura 1.12: Acquisizione con filtro rosso.

Dopo una attenta analisi radiometrica delle immagini si è potuto notare come la

¹Tale inconveniente verrà sfruttato per migliorare i tempi di individuazione

luminosità dei fari posteriori di un veicolo sia rimasta praticamente invariata mentre i riflessi dei catarifrangenti della Figura 1.12 siano molto attenuati rispetto a quelli della Figura 1.11.

In realtà, anche l'intensità luminosa dei fari posteriori risulta leggermente inferiore. Tale svantaggio però non deve trarre in inganno. Come accennato nel paragrafo precedente, non sarà possibile determinare con certezza la presenza o assenza di un veicolo solo dallo studio di un singolo *frame*. L'attenuazione in maniera sensibile i disturbi creati da fonti luminose diverse dai veicoli, migliora il riconoscimento attraverso l'elaborazione di un numero inferiore di immagini. Tale condizione permette non solo di avere un grado maggiore di confidenza sul risultato, ma anche di ottenere in minor tempo l'informazione che pilota i fari; data la natura del problema, il tempo di risposta del sistema risulta di importanza fondamentale.

1.3 Tecniche possibili per la risoluzione del problema

In questo paragrafo parleremo di quali approcci sono già stati sperimentati e quali invece non sono ancora stati progettati per il riconoscimento di un veicolo in un ambiente stradale strutturato.

Innanzitutto, lavorando nell'ambito di un progetto commissionato dal Centro Ricerche Fiat, abbiamo avuto precise specifiche da rispettare. Queste specifiche impongono l'uso della telecamera a toni di grigio descritta nell'Appendice A, montata internamente al veicolo e l'individuazione di tutti i veicoli compresi i motocicli. Tali vincoli, dettati anche da fini commerciali, impongono non poche restrizioni alle diverse possibilità di risolvere il problema posto. Una telecamera a colori avrebbe sicuramente semplificato il problema [3] ma con tutta probabilità non avrebbe avuto un rapporto costi-prestazioni compatibili con le finalità del progetto. Oppure, sarebbe stato interessante condurre uno studio di fattibilità del problema usando una telecamera ad infrarossi. Essendo i veicoli a motore mezzi molto caldi e di dimensioni notevoli rispetto ad un paletto o ad un pedone, forse sarebbe stato possibile individuare veicoli in tempi inferiori e con maggiori prestazioni sulla distanza di localizzazione [1] [2]. Anche se tale metodo avrebbe parzialmente svincolato il problema dalle condi-

zioni ambientali, non avrebbe permesso di distinguere una macchina parcheggiata da poco a fari spenti e un veicolo fermo ad un incrocio semaforico. Inoltre, per garantire una localizzazione di un veicolo ad una distanza dell'ordine del centinaio di metri, occorrerebbero telecamere ad infrarossi aventi prezzi elevatissimi.

La scelta di utilizzare una telecamera a toni di grigio, risulta essere comunque un ottimo compromesso fra prestazioni e prezzo. Infatti, usando metodi di visione artificiali su immagini di questo tipo, è possibile non solo individuare veicoli, ma tanti altri elementi, quali ad esempio le linee della carreggiata e gli ostacoli in prossimità, e sfruttare lo stesso sistema *hardware* per generare altre funzioni utili al controllo automatico ed intelligente del veicolo.

Esiste qualche lavoro già commercializzato su veicoli di serie che riguarda il controllo dei fari abbaglianti. Per controllo si intende solamente l'oscuramento dei fari abbaglianti. In questi veicoli vengono montati speciali sensori chiamati *flash-to-pass* in grado di riconoscere un abbagliamento istantaneo. Quando il sensore intercetta questo *flash* di luce, comanda un motore elettrico che abbassa una piccola lastra in modo da oscurare il fascio di luce degli abbaglianti. Ovviamente tale metodo implica, se pur istantaneamente, un abbagliamento da parte del veicolo che percorre la strada nel senso di marcia opposto. Tale approccio risulta quindi molto limitato: il sensore, infatti, non rivela la presenza di un veicolo se questi non accende gli abbaglianti ed inoltre non agisce se segue da vicino un veicolo nello stesso senso di marcia.

Una volta visionati i filmati fornitici dalla Divisione Tecnologie Ottica e Fotonica del C.R.F. e fatte le considerazioni di cui sopra, il problema si è concentrato sui fattori di distinzione tra i veicoli e gli altri oggetti presenti nelle immagini. Dato il tipo di camera risulta impossibile, anche se nelle immediate vicinanze, cercare di individuare la carrozzeria esterna del veicolo. Inoltre, avendo una sola camera e non una coppia stereo, ogni tentativo di stimare la velocità del veicolo incrociato, pur conoscendo la velocità con la quale la vettura sperimentale si sta muovendo, diventa completamente impreciso: non è possibile riconoscere un veicolo dalla sua velocità con i mezzi a disposizione. Per tutte queste ragioni, la ricerca di automobili e motocicli si è basata sullo studio delle intensità dei fari e del numero di volte in cui compaiono singolarmente in una sequenza di immagini, unita all'andamento della strada. Dato che una

automobile ha una coppia di fari allineati e simmetrici, sono state sperimentate anche funzioni basate sull'individuazione di coppie di fari ottenendo risultati soddisfacenti. Lo svantaggio di queste funzioni è che non possono garantire il riconoscimento dei motocicli dato che possiedono un solo faro posteriore o anteriore. Tale svantaggio risulta essere non ammissibile per l'applicazione commerciale dell'algoritmo.

Capitolo 2

Preelaborazione delle immagini in ingresso

Per preelaborazione si intendono tutti quei metodi che consentono di adattare le immagini acquisite in modo da rendere più efficiente il riconoscimento di un veicolo. In questo capitolo vengono discussi tre di questi metodi, direttamente ideati e sperimentati sulle immagini acquisite staticamente con veicoli inquadrati a distanza nota e sui filmati a noi pervenuti dal C.R.F. .

Il risultato finale della preelaborazione sono due immagini differenti ricavate dall'elaborazione di una particolare zona del *frame* che chiameremo regione d'interesse. Tale regione è raffigurata dal rettangolo di colore verde nella Figura 2.1. e ha dimensione 640 per 150 pixel. La posizione e le dimensioni sono state ricavate dopo aver esaminato attentamente tutti i filmati in nostro possesso, tenendo conto dei possibili percorsi stradali e della calibrazione della camera.

Fondamentalmente, restringendo il campo della preelaborazione a questa finestra, si scartano con sicurezza elementi che potrebbero generare errori nell'interpretazione, compresi i riflessi generati dal cofano del veicolo, e si riduce notevolmente il tempo di esecuzione del programma. È da notare che, considerando anche la pendenza della strada e il beccheggio del veicolo, all'esterno di questa zona non compaiono mai fari di veicoli.



Figura 2.1: Regione dell'immagine da preelaborare.

L'immagine in Figura 2.1 è stata acquisita staticamente con fari abbaglianti spenti e verrà utilizzata come esempio per illustrare la fase di preelaborazione. I veicoli inquadrati sono in posizioni note: quello di sinistra dista 50 metri, mentre quello di destra dista 100 metri. Ora, segue la considerazione di due differenti metodi di preelaborazione e il metodo finale come unione dei due.

2.1 Metodo della doppia Binarizzazione

Questo metodo è stato utilizzato per la prima prova dimostrativa del funzionamento del software di riconoscimento veicoli tenuta presso la Facoltà di Ingegneria di Parma il 13 luglio 2004 con lo staff del Centro Ricerche Fiat.

Semplicemente, converte l'immagine in ingresso da dieci bit a otto bit utilizzando un centro di conversione fissato e ricava due immagini binarizzate secondo due diverse soglie fissate a priori. Questo metodo sfrutta il fatto che l'intensità dei fari posteriori è tipicamente inferiore a quella dei fari anteriori, per cui l'immagine binarizzata con soglia alta contiene i fasci luminosi dei fari anteriori, mentre con quella con soglia bassa contiene sia i fasci alti che quelli bassi e illuminazione stradale. Con una opportuna funzione, vengono individuate le posizioni dei fasci molto luminosi e cancellati nell'immagine binarizzata con soglia bassa così da distinguere completamente l'analisi per fari anteriori da quelli posteriori.

Tale algoritmo è veloce computazionalmente e permette di riconoscere anche un veicolo nello stesso senso di marcia a 150 metri. Inoltre, nell'immagine prodotta dalla binarizzazione alta sono presenti pochi elementi di rumore. Questa tecnica, però, presenta i seguenti svantaggi:

- Dato che le soglie di binarizzazione sono costanti, una variazione anche lieve delle condizioni di luminosità dell'ambiente potrebbe risultare molto critica per l'algoritmo, provocando l'individuazione di falsi positivi.
- L'immagine binarizzata con soglia bassa risulta essere molto rumorosa, nonostante sia applicato all'obiettivo della telecamera un filtro rosso.

2.2 Metodo della Differenza

Il grande svantaggio del metodo precedente è la troppa sensibilità alla variazione della luminosità esterna o ambientale. Il secondo metodo che viene ora illustrato risolve questo problema. Prima di spiegarne l'algoritmo è opportuno dire che al seguito di un attento esame delle immagini acquisite con filtro rosso, risulta che il valore dell'intensità dei fari si concentra quasi esclusivamente nella metà alta delle 1024 tonalità di grigio possibili. Questo permette di eliminare ulteriormente informazioni prive di significato come parte dell'illuminazione esterna, linee di demarcazione della corsia e cartelli.

Nella Figura 2.2 viene rappresentato l'istogramma dell'immagine in Figura 2.1 . In colore rosso viene illustrato l'istogramma calcolato sull'intera immagine. Sulle ascisse vengono rappresentati i 1024 toni di grigio possibili, mentre sulle ordinate, il numero di pixel (massimo 478) corrispondenti. Il rettangolo verde inquadra tutti quei valori di grigio fino a 520 che non sono utili per rappresentare i fari dei veicoli. Tutti i pixel aventi valore di grigio inferiore o uguale a 520 verranno posti uguale a zero. Se i fari abbaglianti sono accesi, però, la luminosità media dell'immagine acquisita aumenta e quindi sarà necessario variare questa soglia. Nel caso di abbaglianti accesi, la soglia viene posta a 730. La scelta di queste soglie come pure della dimensione

del filtro di media, sono il frutto di una lunga sperimentazione applicata non solo alle immagine acquisite staticamente, ma anche a tutti i filmati forniti dal C.R.F.

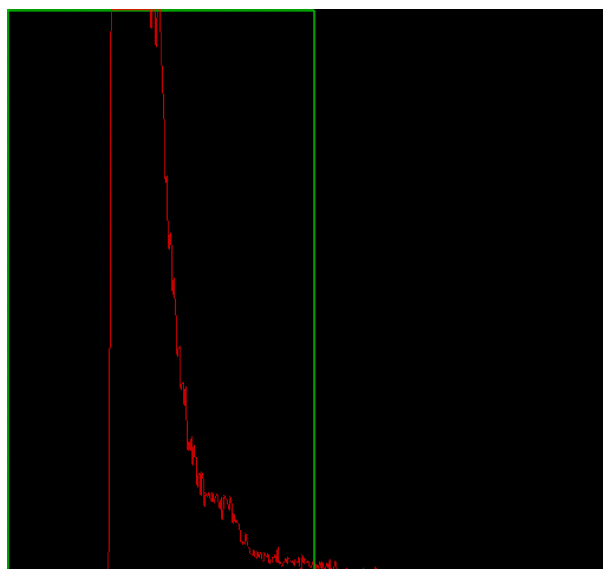


Figura 2.2: Istogramma (colore rosso) della Fig. 2.1.

Onde studiare in maniera più approfondita le caratteristiche radiometriche di una immagine, si è creato un altro tipo di istogramma che ha come ascisse la somma dei valori di intensità di tutti i pixel per una medesima colonna e come ordinata le 640 colonne dell'immagine. Il risultato che si ottiene è quello in Figura 2.3.

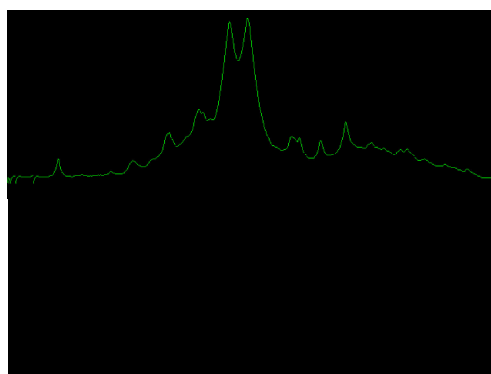


Figura 2.3: Istogramma delle intensità (colore verde) della Figura 2.1.

Come si può notare in Figura 2.3, i picchi più rilevanti dell'istogramma sono in

corrispondenza dei fari delle automobili e dai catarifrangenti più vicini. Da qui nasce l'idea principale del secondo metodo: sottrarre l'immagine originale a dieci bit con il risultato dell'applicazione di un filtro passa basso o filtro di media. Le operazioni di differenza e di filtraggio vengono applicate solo alle regioni d'interesse. La particolarità del filtro di media è quella di tendere a diminuire i picchi e aumentare i minimi locali dell'istogramma. Proporzionalmente all'aumentare della derivata del picco, il filtro ne diminuisce il valore massimo. Il risultato di questa operazione dipende da come il filtro viene implementato e da quante volte viene ricorsivamente applicato all'immagine. In particolare, l'algoritmo assegna ad ogni pixel il valore medio dei venticinque pixel circostanti e lo itera per tre volte consecutivamente a tutta la regione d'interesse. Come accennato, però, la riflessione dei catarifrangenti con abbaglianti accesi è superiore che nel caso di abbaglianti spenti. Per cui, nell'elaborazione con abbaglianti accesi viene iterato lo stesso tipo di filtro non più per tre volte ma per sei. Il fatto che l'illuminazione media di una immagine acquisita ad abbaglianti accesi sia superiore, non influisce sull'elaborazione.

Confrontando i due istogrammi intensità dell'immagine originale e la stessa mediata si ottiene questa rappresentazione, facendo distinzione fra acquisizione con abbaglianti spenti o meno:

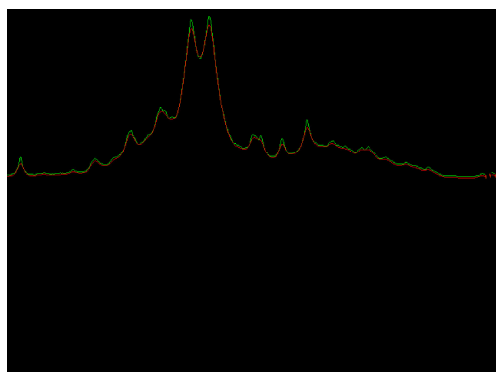


Figura 2.4: Confronto fra istogrammi dopo 3 iterazioni del filtro di media.

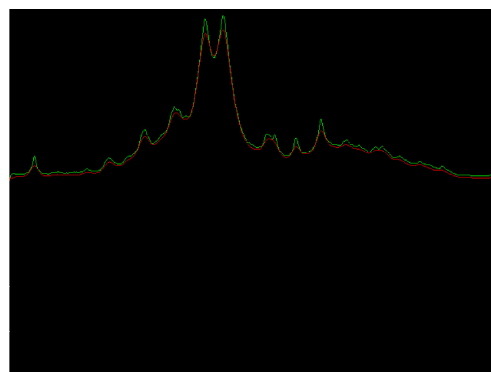


Figura 2.5: Confronto fra istogrammi dopo 6 iterazioni del filtro di media.

L'istogramma rosso è stato ottenuto dall'immagine filtrata. Si nota che è proprio in corrispondenza dei picchi che la differenza fra i due istogrammi aumenta. I con-

fronti fra gli istogrammi riportati nelle figure sopra sono stati ricavati dalla medesima immagine di partenza ma iterando per un numero diverso il filtro di media. Aumentando il numero di iterazioni del filtro di media, aumenta la differenza fra l'istogramma dell'immagine originale e quella filtrata.

Facendo la differenza, quindi, il valore dell'illuminazione ambientale non influisce sul risultato. Per descrivere l'immagine differenza senza perdere informazioni, sono sufficienti 8 bit. Sottraendo così l'immagine originale con quella mediata si ottengono solo gli oggetti più luminosi.



Figura 2.6: Immagine differenza.

Dalla Figura 2.6, si vede come compaiano almeno due elementi di disturbo sul lato destro della carreggiata. La conoscenza della posizione della carreggiata aiuterebbe ad eliminare questi elementi. Questa idea è stata sviluppata nell'algoritmo. Parallelamente alla fase di preelaborazione viene calcolato l'andamento della strada e questa informazione viene passata alla funzione che filtra la regione d'interesse, evitando di mediare la zona all'infuori della carreggiata, così da ottenere un'immagine differenza senza i due catarifrangenti sul lato destro. Questa fase, però, sarà spiegata in maniera molto più approfondita nel Capitolo 3.

I vantaggi di questo metodo sono:

- insensibilità alla variazione di luminosità ambientale.
- buone prestazioni sull'eliminazione di tutto ciò che è rumore e relativo aumento dell'affidabilità dei risultati ottenuti.

Comunque, rimane la difficoltà nel rilevare macchine nello stesso senso di marcia a distanze elevate. Inoltre il carico computazionale è elevato, ma sostenibile per la sperimentazione in tempo reale.

Lo svantaggio principale è la mancata rilevazione di veicoli nel senso di marcia opposto con o senza abbaglianti accesi relativamente vicini (meno di 20 metri). Con questo metodo vengono rilevati solo i bordi degli oggetti luminosi di grosse dimensioni. Questo svantaggio è dovuto al problema di mediare in zone molto chiare estese: mediando in tali zone il valore dell'intensità rimane alto e sottraendo con la corrispondente zona originale si ottiene una zona scura.

Questi effetti sono evidenti nella Figura 2.7.

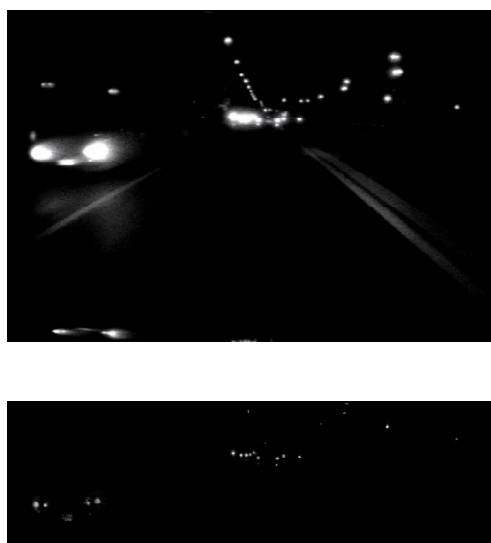


Figura 2.7: Estrapolazione dell'immagine differenza.

Per i motivi spiegati sopra, i fari del veicolo in basso a sinistra della Figura 2.7 non sono ben definiti, ma compaiono per lo più i bordi. Questo causa una mancata rilevazione dei fari da parte delle fasi successive dell'algoritmo.

L'algoritmo termina con la binarizzazione dell'immagine differenza con una soglia fissa: l'immagine così ottenuta è pronta per essere sottoposta alle procedure di riconoscimento dei fari.

2.3 Unione dei metodi di Binarizzazione e Differenza

Questo metodo è l'unione dei metodi precedenti: viene utilizzata una immagine binarizzata con soglia molto elevata per determinare le macchine con abbaglianti acce-

si o relativamente vicine e la restante illuminazione viene mantenuta dall'immagine differenza.

Mentre, l'immagine differenza viene ottenuta come spiegato nel paragrafo precedente, l'altra immagine viene ricavata da una binarizzazione con soglia fissa della regione d'interesse (vedi Figura 2.1) dell'immagine originale convertita ad otto bit. Siccome tutti i valori di grigio relativi ai fari sono esclusivamente nella parte alta dei 1024 possibili, la conversione viene centrata utilizzando un valore predefinito alto: questo permette non solo di eliminare una buona parte di rumore, ma anche di non perdere informazioni sui fari. Questa immagine viene ricavata molto più velocemente di quella differenza.

A volte i due metodi rivelano le sagome degli stessi fari: questo viene sfruttato in una logica di possibile conferma. In tal modo si ottengono tutti i vantaggi del metodo sottrazione e la soluzione allo svantaggio principale con il metodo binarizzazione. I vantaggi di questo metodo sono:

- insensibilità alla variazione di luminosità ambientale.
- buone prestazioni sull'eliminazione di tutto ciò che è rumore e relativo aumento dell'affidabilità dei risultati ottenuti.
- l'immagine binarizzata con soglia alta non contiene mai rumore.
- riconoscimento sicuro e immediato di un veicolo se presente nell'immagine binarizzata con soglia alta.
- aumento delle prestazioni in termini di tempo e distanza di localizzazione di un veicolo

Resta comunque difficile individuare macchine nello stesso senso di marcia a distanze molto elevate. Inoltre il tempo di elaborazione di un *frame* rimane elevato. Questo non rappresenta un problema dell'algoritmo in quanto il tempo impiegato ad elaborare una immagine è inferiore all'intervallo di tempo di acquisizione; tuttavia potrebbe risultare un limite al numero massimo di frame acquisiti in un secondo.

Il risultato della preelaborazione dell'immagine in Figura 2.1 sono due immagini binarizzate di dimensioni 640x150 pixel come in Figura 2.8 e 2.9 .



Figura 2.8: Immagine binarizzata con soglia alta.

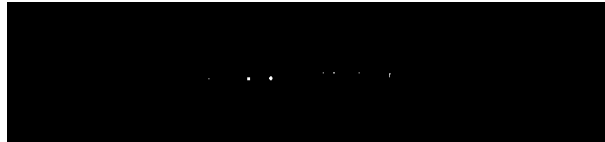


Figura 2.9: Immagine differenza binarizzata.

In Figura 2.9 sono visibili i fari posteriori del veicolo di destra, mentre nella Figura 2.8 compaiono solo i fari anteriori del veicolo di sinistra.

Capitolo 3

Individuazione dell'andamento della strada

Per aumentare l'affidabilità della rilevazione di un veicolo si è reso necessario introdurre la conoscenza dell'andamento della strada. In particolare l'algoritmo è in grado di stimare la posizione della strada dall'analisi dell'illuminazione stradale verticale e dalle linee di demarcazione della carreggiata [5] [6]. Per far questo si applica ad opportune regioni dell'immagine, una particolare trasformata matematica, chiamata trasformata di Hough. Quando l'algoritmo riconosce la strada elimina una porzione di immagine relativa al lato destro della carreggiata. Questo permette di togliere numerosi elementi di "disturbo" nell'immagine. L'introduzione della conoscenza della posizione della carreggiata, ha notevolmente migliorato le prestazioni nel corretto riconoscimento dei veicoli.

3.1 Uso della trasformata di Hough

La trasformata di Hough è una tecnica che permette il riconoscimento di configurazioni globali presenti in una immagine (segmenti, curve, forme prestabilite), e si basa sulla trasformazione di tutti i punti costituenti una immagine, in punti di un nuovo spazio detto spazio dei parametri. Nel nostro progetto la trasformata di Hough si applica ad immagini binarie, ovvero immagini in cui sono presenti due soli livelli, bianco e

nero, ed in cui l'informazione associata ad un punto è rappresentata unicamente dalla sua posizione. Questa tecnica, ideata da Hough nel 1962, è basata sulla "validazione delle ipotesi" in cui, definita la curva che si intende cercare nella scena, per ogni punto dell'immagine si calcolano i parametri di tutte le curve che potrebbero passare per quel punto e si incrementano le celle di uno spazio n-dimensionale (con n numero dei parametri) che corrispondono alle varie curve. Si ottiene così una funzione di accumulazione definita nello spazio dei parametri. Alla fine saranno i massimi di questa funzione, ovvero i punti nello spazio dei parametri che hanno accumulato il maggior numero di "voti" a rappresentare le curve che hanno probabilità elevata di essere presenti nell'immagine, come se si trattasse di una ipotesi avvalorata dal maggior numero di conferme sperimentali.

Una delle caratteristiche più interessanti della Trasformata di Hough è quella di non risentire del rumore presente in una immagine. Infatti, l'altezza dei picchi nella matrice di accumulazione, dipende in maniera del tutto trascurabile da eventuali lacune presenti nella retta di partenza o dall'esistenza di punti spuri presenti nello spazio immagine.

3.1.1 Ricerca di rette

La retta può essere parametrizzata in molti modi. Un modo conveniente è quello di scegliere come parametri la distanza ρ della retta dall'origine e l'angolo θ formato dalla normale alla retta con l'asse delle ascisse (forma normale di Hesse): in questo caso l'equazione parametrica della retta è

$$\rho = x \cos \theta + y \sin \theta \quad (3.1)$$

Così facendo entrambi i parametri risultano illimitati, a differenza dei parametri a e q della forma

$$y = ax + q \quad (3.2)$$

che non sono limitati (a è infinito per rette verticali).

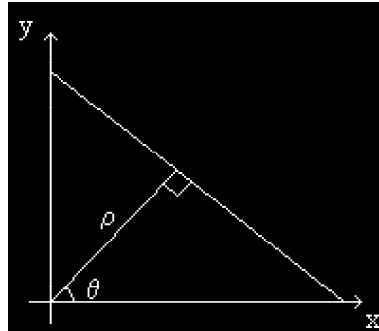


Figura 3.1: Linea parametrica.

Per sapere quale curva corrisponde nello spazio dei parametri al singolo punto dell'immagine si devono considerare tutte le rette passanti per il punto stesso, ossia il fascio di rette con centro nel punto considerato. Allora, dato che tutte queste rette hanno in comune il centro, che si impone avere coordinate x_0 e y_0 , la trasformata del punto sarà la curva di equazione

$$\rho = x_0 \cos \theta + y_0 \sin \theta \quad (3.3)$$

ossia una sinusoidale nel piano $\rho - \theta$. La trasformata dell'intera immagine sarà invece la sovrapposizione di tante sinusoidi quanti sono i punti che compongono l'immagine stessa. Se alcuni di questi punti sono attraversati da una stessa retta, ossia sono allineati, le corrispondenti sinusoidi si incroceranno in uno stesso punto del piano dei parametri le cui coordinate sono i parametri che caratterizzano la retta che passa per quei punti.

La trasformata di Hough va quantizzata per essere memorizzata in una matrice. Pertanto anziché una trasformazione vera e propria si ottiene una sorta di matrice di accumulazione, detta mappa di Hough, in cui ogni cella contiene il numero di sinusoidi che passano nell'intervallo di quantizzazione associato alla cella stessa: le celle che contengono i valori di accumulazione più elevati rivelano potenziali forme all'interno dell'immagine che meglio di altre assomigliano a rette.

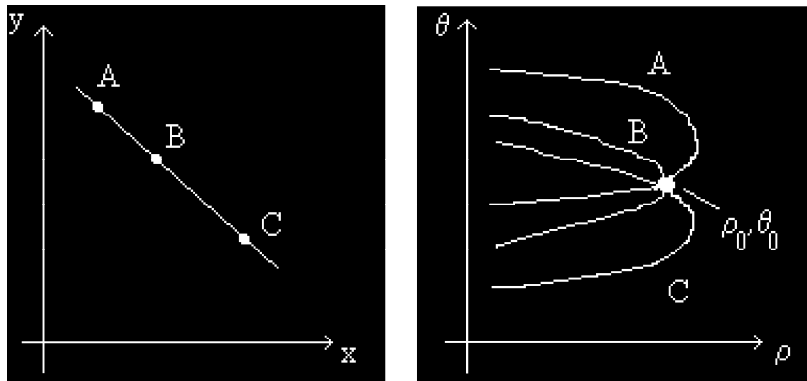


Figura 3.2: Tre punti allineati. Figura 3.3: Trasformata di Hough.

3.1.2 Illuminazione stradale

Come accennato nell'introduzione a questo capitolo, la trasformata è insensibile ad eventuali lacune nella retta da individuare, in quanto i valori massimi dello spazio accumulatore si ottengono per somma di sinusoidi rappresentanti le possibili rette per ogni punto nell'immagine. Questa procedura funzionerebbe benissimo anche con solo due o tre punti allineati. Pertanto, dato che l'illuminazione verticale in un ambiente stradale strutturato ha un andamento lineare rivolto verso il basso per effetto della prospettiva, è possibile ricavare il punto di fuga dell'immagine con una precisione sufficiente per determinare la posizione della carreggiata.

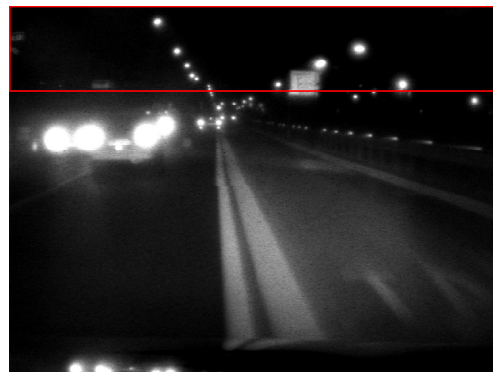


Figura 3.4: Porzione d'immagine sulla quale viene applicata la trasformata di Hough.

Il rettangolo rosso in Figura 3.4 circonda la porzione di immagine sulla quale

verranno eseguite le procedure di rilevamento della strada. Le dimensioni e la posizione restano costanti. A questa immagine viene applicato un filtro per individuare i contorni degli oggetti presenti. Le procedure di *Edge Detection* sono ampiamente documentate in letteratura e ne esistono numerose varianti. Nel nostro caso, per determinare se un pixel fa parte di un contorno, si esamina il valore assoluto della differenza radiometrica con gli otto pixel circostanti: se almeno una di queste differenze è maggiore di una determinata soglia, allora il pixel esaminato fa parte di un contorno e viene posto a valore massimo. Se nessuna differenza è maggiore della soglia allora il pixel viene posto a zero. Queste operazioni si iterano per ogni pixel bianco dell'immagine, ottenendo il risultato in Figura 3.5.



Figura 3.5: Contorni dell'illuminazione stradale.

Viene eseguita la trasformata di Hough sull'immagine binaria ricavata. Nella implementazione di Hough da noi creata tutte le rette possono essere descritte con un angolo θ compreso fra zero e π .

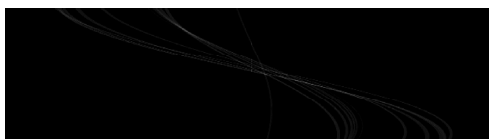


Figura 3.6: Spazio accumulatore.

Nella Figura 3.6 viene visualizzato lo spazio accumulatore creato dalla trasformata. Gli assi ρ e θ sono orientati come in Figura 3.3. Ovviamente, è stato necessario individuare nello spazio accumulatore tutti i punti con valori elevati ed estrapolarne i corrispondenti valori di ρ e θ . Ad esempio, preso il punto P di coordinate $(\rho_0$ e $\theta_0)$ è necessario applicare una trasformazione per ottenere i valori corretti di ρ e θ relativi alla retta individuata dal P. La trasformazione che si impone è:

$$\rho = \rho_0 - \frac{H_{width}}{2} \quad (3.4)$$

$$\theta = \frac{(\theta_0 * \pi)}{H_{height}} \quad (3.5)$$

dove H_{width} e H_{height} sono la larghezza e l'altezza della spazio accumulatore.

Una volta estrapolati questi due parametri, vengono scartate tutte le rette che non soddisfano la seguente proprietà:

$$\theta \leq \frac{3 * \pi}{8} \cup \theta > \frac{5 * \pi}{8} \quad (3.6)$$

Questo alleggerisce il carico computazionale scartando tutte quelle rette la cui pendenza non è compatibile con quella delle linee generate dall'illuminazione stradale.

Successivamente si trasformano le coordinate ricavate in forma normale di Hesse in coordinate cartesiane. Andando a graficare le rette trovate nell'immagine di partenza si ottengono i risultati mostrati in Figura 3.7.

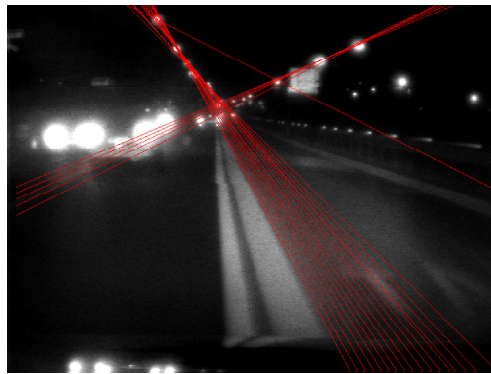


Figura 3.7: Rappresentazione delle linee individuate.

Per determinare il punto di fuga, l'algoritmo cerca le intersezioni fra le rette. Non sono considerate intersezioni valide quelle tra le rette i cui coefficienti angolari differiscono almeno di $\frac{\pi}{4}$. Questo permette di evitare intersezioni fra linee generate dal medesimo contorno. Date due rette, $R_1(\rho_1, \theta_1)$ e $R_2(\rho_2, \theta_2)$, l'intersezione viene determinata dal seguente sistema:

$$\begin{cases} \rho_1 = x_0 \cos \theta_1 + y_0 \sin \theta_1 \\ \rho_2 = x_0 \cos \theta_2 + y_0 \sin \theta_2 \end{cases} \quad (3.7)$$

Risolvendo il sistema si ricavano le coordinate del punto intersezione.

$$\begin{cases} x_0 = \frac{\rho_2 - y_0 \sin \theta_2}{\cos \theta_2} \\ y_0 = \frac{\rho_1 - \left(\frac{\rho_2 \cos \theta_1}{\cos \theta_2} \right)}{\frac{\sin \theta_1}{1 - (\cos \theta_1 \sin \theta_2)}} \frac{1}{\cos \theta_2 \sin \theta_1} \end{cases} \quad (3.8)$$

La posizione del punto di fuga è determinata dalla posizione del pixel intersezione che ha il numero maggiore di pixel intersezione nel suo intorno. L'intorno ha dimensione 400 pixel. Se non esiste una intersezione con almeno altre quattro nel suo intorno, il punto di fuga non è determinato. Questo permette di ottenere un risultato sufficientemente preciso e affidabile.

Le coordinate del punto di fuga permettono di creare una linea spezzata che rappresenta il confine al di sopra del quale gli oggetti luminosi presenti vengono scartati in fase di preelaborazione. La funzione che descrive la linea di confine è riportata nella formula 3.9.

$$\begin{cases} y = y_0 + 5 & x < x_0 + 100 \\ y = 2y_0 + \frac{x_0 y_0}{100} - \frac{y_0}{100} x & x \geq x_0 + 100 \end{cases} \quad (3.9)$$

Dove le coordinate del punto di fuga sono x_0 e y_0 .

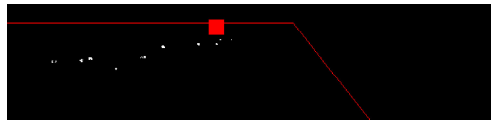


Figura 3.8: Limitazione della regione d'interesse.

Il punto di fuga è rappresentato dal centro del quadrato rosso in Figura 3.8. La posizione della linea rossa è calcolata in base alla posizione del punto di fuga. Come si nota in Figura 3.8 parte dell'illuminazione verticale e catarifrangenti a lato della strada vengono scartati. In assenza dell'informazione della posizione della carreggiata sarebbero stati considerati.

3.1.3 Linee della carreggiata

Il procedimento per determinare le linee della carreggiata è del tutto analogo a quello dell'illuminazione stradale verticale. Ovviamente, varia la porzione dell'immagine che viene processata.



Figura 3.9: Porzione d'immagine sulla quale viene applicata la trasformata di Hough.

In questo caso, però, mentre l'illuminazione stradale rimane costante se variano leggermente le condizioni di illuminazione stradale, le linee della carreggiata, dipendono fortemente dall'illuminazione dei fari del veicolo. Sarà quindi necessario variare la soglia di rilevazione dei contorni nel caso di abbaglianti spenti oppure accesi. Tali situazioni sono già state testate ottenendo buoni risultati in entrambi i casi.

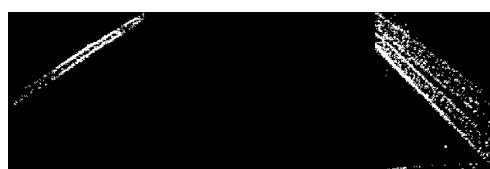


Figura 3.10: Contorni delle linee della carreggiata.

Nella Figura 3.10 è possibile notare come l'estrazione dei contorni generi un'immagine con diversi punti bianchi di rumore che non appesantirebbero computazionalmente l'algoritmo, ma non appartengono alle linee stradali. Al fine di evitare ciò si applica una funzione di erosione all'immagine. L'implementazione di questa funzione è molto semplice: per ogni pixel bianco si calcolano quanti pixel bianchi sono presenti nell'intorno di dimensione 64 pixel. Se ci sono meno di sedici pixel bianchi significa

che il punto non fa parte del contorno e viene posto uguale a zero, altrimenti, il punto appartiene ad un contorno e non viene fatta nessuna altra operazione.



Figura 3.11: Contorni delle linee della carreggiata dopo l'erosione.

Dall'immagine in Figura 3.11 si ricava lo spazio accumulatore in Figura 3.12.



Figura 3.12: Spazio accumulatore.

Una volta ottenuti, i parametri ρ e θ dei valori massimi dello spazio accumulatore, si applica la trasformazione dalle coordinate in forma normale di Hesse in coordinate cartesiane. Andando a disegnare le rette trovate nell'immagine di partenza si ottengono i risultati che seguono.

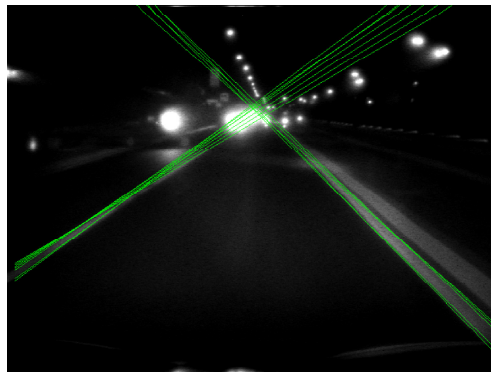


Figura 3.13: Rappresentazione delle linee individuate.

Successivamente, con le medesime modalità e parametri dell'algoritmo applicato all'illuminazione stradale, viene delimitata la regione d'interesse che servirà per la ricerca dei possibili fari.

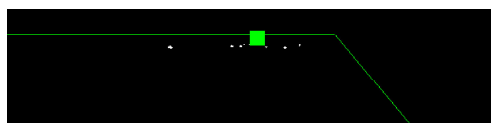


Figura 3.14: Limitazione della regione d'interesse.

Il punto di fuga è rappresentato dal centro del quadrato verde in Figura 3.14. La posizione della linea verde è calcolata in base alla posizione del punto di fuga. Come si nota in Figura 3.14 parte dell'illuminazione verticale e catarifrangenti a lato della strada vengono scartati. In assenza dell'informazione della posizione della carreggiata sarebbero stati considerati.

3.2 Considerazioni sull'applicabilità

I vantaggi di introdurre la conoscenza del punto di fuga e la relativa posizione della carreggiata sono di eliminare diverso rumore dall'immagine. Dalle prove sperimentali è emerso che sia i punti di fuga individuati dall'illuminazione stradale e dalle linee sulla carreggiata, quando è stato possibile calcolarli entrambi, coincidono. Tale risultato permette, non solo di considerare il punto trovato sufficientemente attendibile, ma anche di ricercarlo soltanto dalle linee della carreggiata o dall'illuminazione stradale. Se il primo metodo fallisce si esegue il secondo: questo permette di ottimizzare i tempi di esecuzione.

Lo svantaggio della ricerca della posizione della carreggiata è che impiega un tempo considerevole. Tale svantaggio, però, non crea problemi in quanto, come sarà evidenziato nel Capitolo 6, il tempo che intercorre fra un frame e l'altro è sufficiente per una elaborazione in tempo reale.

La defezione principale sta nel fatto che non sempre è presente illuminazione verticale e linee di demarcazione della carreggiata. Tutto ciò, però, non dipende dall'implementazione dell'algoritmo ma da quanto è strutturato l'ambiente stradale. Lo stesso stato delle linee stradali ne influenza la rilevazione.

Inoltre, la riflessione delle linee orizzontali dipende da quanto sono illuminate dai fari del veicolo. Ad esempio, nell'immagine in Figura 3.15, l'algoritmo non riesce a

calcolare il punto di fuga perchè non è presente illuminazione verticale e la linea destra della carreggiata è poco illuminata dal faro destro.



Figura 3.15: Porzione d'immagine sulla quale viene applicata la trasformata di Hough.

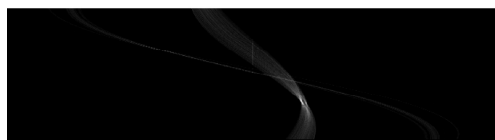


Figura 3.16: Spazio Accumulatore.



Figura 3.17: Contorni delle linee della carreggiata.

L'algoritmo, riconosce il tratto di linea di sinistra ma non quello di destra per cui non ci sono punti d'intersezione.

Una nota particolare riguarda il riconoscimento della strada percorrendo una curva. L'algoritmo non è in grado di riconoscere curve aventi un raggio di curvatura accentuato. Questo perchè le linee sul percorso stradale non sono rettilinee ma curve e l'illuminazione verticale è parzialmente presente nell'immagine. Anche se questo rappresenta una mancanza, non risulta determinante nelle prestazioni dell'algoritmo. Per capire il perchè si rimanda al Capitolo 5 che spiega il comportamento dell'algoritmo di *tracking dei fari* nel caso in cui il veicolo sperimentale percorra una curva.

3.2.1 Prestazioni del rilevamento della posizione della carreggiata

Riportiamo ora una serie di tabelle che schematizzano quanto sia efficace l'algoritmo illustrato in questo Capitolo. Le Tabelle 3.1 e 3.2 riportano i risultati ottenuti elaborando sequenze acquisite esclusivamente con fari abbaglianti spenti. Pertanto l'algoritmo ha mantenuto costanti le impostazioni di elaborazione ad abbaglianti spenti.

Sequenza	Num. tot. immagini	% rilevazione
RedFilter_M1	2516	21
RedFilter_M2	2650	29
RedFilter_M3	7315	15

Tabella 3.1: Prestazioni in percentuale sull'individuazione della carreggiata con abbaglianti spenti.

Sequenza	Num. tot. immagini	Dalle linee carreggiata	Da illum. stradale
RedFilter_M1	2516	482	98
RedFilter_M2	2650	762	2
RedFilter_M3	7315	895	206

Tabella 3.2: Prestazioni con abbaglianti spenti.

I valori ottenuti dipendono fortemente da come è strutturato il percorso stradale. Ad esempio la sequenza *RedFilter_M2* è stata acquisita percorrendo un percorso quasi totalmente privo di segnaletica verticale, pertanto si spiega il numero esiguo di volte che l'algoritmo ha individuato la posizione della carreggiata elaborando la porzione di immagine contenente illuminazione stradale.

Di seguito vengono riportate le prestazioni dell'elaborazione di sequenze acquisite con fari abbaglianti accesi. Si nota che l'algoritmo ha mantenuto costanti le impostazioni di elaborazione ad abbaglianti accesi. I risultati ottenuti sono più scarsi di quello che si ci poteva aspettare. Questo è dovuto alla leggera variazione del cablaggio della camera sul veicolo. Tale variazione influenza non poco l'individuazione delle linee perchè l'algoritmo si basa sull'elaborazione di zone predefinite nell'immagine: una

leggera variazione della calibrazione della camera comporta una significativa alterazione di quello che si aspetta essere presente nelle regioni precalcolate nell'immagine. Questo problema, comunque, è dovuto solo ad un errore di cablaggio e come tale è facilmente risolvibile.

Sequenza	Num. tot. immagini	% rilevazione
RedFilter_S4	1086	32
RedFilter_S5	3328	50
RedFilter_S6	3754	3
RedFilter_S7	7259	18

Tabella 3.3: Prestazioni in percentuale sull'individuazione della carreggiata con abbaglianti accesi.

Sequenza	Num. tot. immagini	Dalle linee carreggiata	Da illum. stradale
RedFilter_S4	1086	345	0
RedFilter_S5	3328	1651	88
RedFilter_S6	3754	83	25
RedFilter_S7	7259	1198	86

Tabella 3.4: Prestazioni con abbaglianti accesi.

È importante prendere in considerazione la possibilità di mantenere, un volta calcolata, l'informazione sulla posizione del punto di fuga per un numero predefinito di immagini consecutive, evitando che l'algoritmo ripeta nuovamente tutta l'elaborazione. Tale eventualità che permette di ottenere tempi di esecuzione sensibilmente più veloci, è stata analizzata ma non applicata. Questo perchè, come si è detto in precedenza, difficilmente l'algoritmo riesce ad individuare correttamente la strada se si sta percorrendo una curva. Se l'algoritmo calcola la posizione della carreggiata percorrendo un rettilineo all'approssimarsi di una curva, potrebbe mantenere il risultato anche quando sta effettivamente percorrendo un curva causando l'eventuale eliminazione di fari effettivamente appartenenti a veicoli. Inoltre, è vero che i tempi di esecuzione medi sarebbero inferiori, ma non aumenterebbero altrettanto sensibilmente le presta-

zioni sulla percentuale di riconoscimenti; dipendendo dalla presenza e dallo stato della segnaletica verticale e delle linee su fondo stradale, i riconoscimenti sono per lo più consecutivi.

In conclusione, la parte dell'algoritmo che si occupa dell'individuazione della posizione della strada è utilizzato come se fosse un sensore virtuale che fornisce, quando possibile, le coordinate del punto di fuga dell'immagine. Quando fornisce un risultato questo è da considerarsi attendibile e di discreta precisione.

Capitolo 4

Caratterizzazione dei fari

Questo capitolo spiega nel dettaglio le procedure utilizzate per il riconoscimento dei fari. Le immagini processate in questa fase sono quelle restituite dalla preelaborazione opportunamente filtrate dalla conoscenza della posizione della carreggiata.

Lo studio della forma dei fasci luminosi dei fari ha portato alla costruzione di modelli generali per la rappresentazione di un faro nell'immagine. E' evidente che non sarà necessario un solo modello in quanto il fascio di luce acquisito dalla camera varia a seconda della distanza del veicolo incrociato, dalla forma del faro, dal tipo di faro e dalle condizioni ambientali. Inoltre, sono necessari più modelli per cercare di distinguere il più possibile un faro da illuminazione stradale o da un paletto catarifrangente a lato della carreggiata

4.1 Modellazione dei fari

Lo studio dei modelli dei fari ha richiesto numerose sperimentazioni. Data la diversità delle immagini prodotte dalla preelaborazione, una dedicata ai veicoli più vicini e quindi con fasci luminosi più grandi e l'altra dedicata ai veicoli lontani, si è reso necessario distinguere i modelli a seconda dell'immagine da processare.

4.1.1 Modelli per immagine binarizzata con soglia alta

Di seguito, si riportano i quattro *pattern* utilizzati (Fig. 4.1).

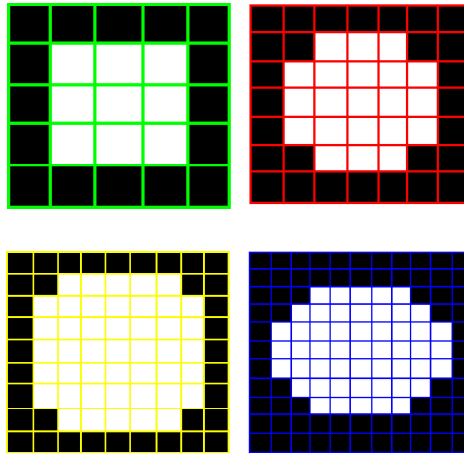


Figura 4.1: Modelli dei fari.

I *pattern* variano non solo per la loro forma ma anche per dimensione. Ognuno ha un bordo di pixel con valore 0. Questo permette di selezionare solo quelle zone non più grandi del modello. In tal modo è possibile dare una dimensione relativamente precisa al faro individuato. Le forme diverse, servono per riconoscere tipologie di fari diverse.

A seconda della dimensione dei modelli con i quali vengono individuati, la posizione del faro viene visualizzata con diversi colori:

- verde dimensione 25 pixel
- rosso dimensione 49 pixel
- giallo dimensione 81 pixel
- blu dimensione 121 pixel

4.1.2 Modelli per immagine differenza

Di seguito, si riportano i quattro *pattern* utilizzati (Fig. 4.2).

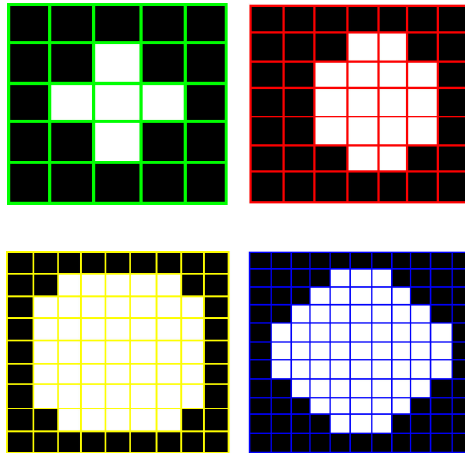


Figura 4.2: Modelli dei fari.

Sia per dimensione che per forma, questi modelli assomigliano a quelli riportati nel paragrafo precedente. Ma nel complesso risultano essere più efficaci nell'individuazione di fasci luminosi molto piccoli.

4.2 Algoritmo di Pattern Matching

La filosofia adottata nella fase di individuazione dei possibili fari è quella di cercare di individuare con precisione tutti gli oggetti aventi una forma pressochè circolare. E' decisamente più corretto localizzare come fari anche illuminazione stradale che mancare di individuare un vero faro nella scena. Dato che i fasci luminosi giungono notevolmente distorti all'obiettivo della camera, l'algoritmo di *pattern matching* individua uno stesso faro anche con più modelli uguali fra loro, ma in posizioni leggermente diverse. Questo permette di individuare sicuramente qualsiasi faro nell'immagine con lo svantaggio di etichettare come fari anche illuminazione stradale.

L'algoritmo di *pattern matching* è molto semplice: scansionando l'intera immagine, ogni qual volta viene individuato un pixel bianco viene confrontato un modello con la corrispondente porzione di immagine centrata sul pixel; se la somma della differenza dei toni di grigio di ogni pixel nella porzione dell'immagine e nel modello è minore di una soglia prefissata allora la porzione di immagine viene etichettata come faro,

salvandone la dimensione, la posizione e di quanto differiva dal modello. La formula utilizzata per il confronto è la seguente:

$$\sum_{i=1}^n |img(i) - pattern_k(i)| < soglia_k \quad (4.1)$$

dove n è la dimensione in pixel del k -esimo modello. L'immagine viene scandita confrontando ogni pixel bianco con tutti e quattro i modelli ripetendo il procedimento appena illustrato. A seconda del modello e dell'immagine processata variano le soglie massime di errore nel confronto.

Di seguito si mostra un esempio di funzionamento in Figura 4.3.



Figura 4.3: Immagine di test.

Nell'immagine in Figura 4.4(a) vengono rilevati 99 fari mentre nell'immagine in Figura 4.4(b) 31. Anche qualitativamente è evidente la ridondanza dei dati. Per eliminare questo esubero di dati è stata implementata una funzione di fusione fra i diversi fari trovati in posizioni molto vicine. Il risultato finale è visibile in Figura 4.4 con rispettivamente quattro e cinque fari selezionati.

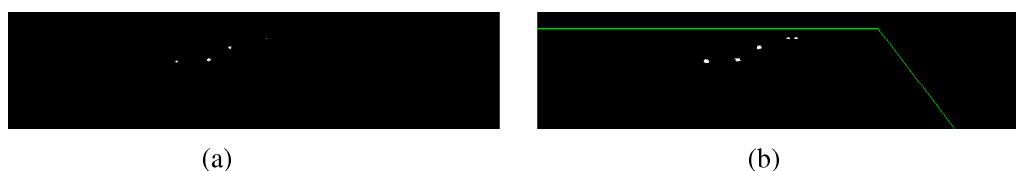


Figura 4.4: (a) Immagine binarizzata con soglia alta, (b) Immagine differenza.

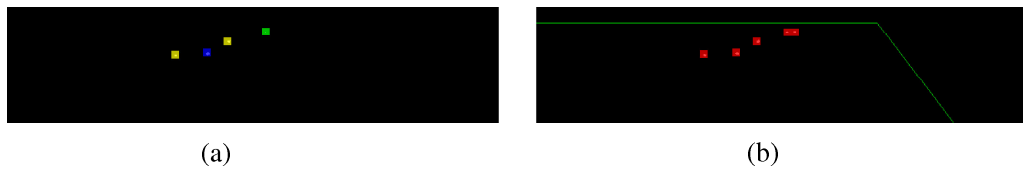


Figura 4.5: (a) Immagine binarizzata con soglia alta, (b) Immagine differenza.

Il colore dei quadrati indica la dimensione del faro. L'algoritmo funziona correttamente e con grande precisione come dimostrano le immagini sopra.

Nel caso in cui i fari del veicolo incrociato, siano vicini e generino fasci di luce grandi, lo stesso faro viene riconosciuto con diversi fari di dimensioni variabili. Tale inconveniente però non costituisce uno svantaggio in quanto in tale posizione, il veicolo è già stato sicuramente individuato dall'algoritmo.

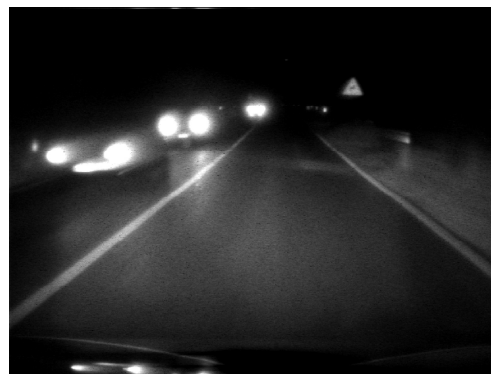


Figura 4.6: Immagine di test.

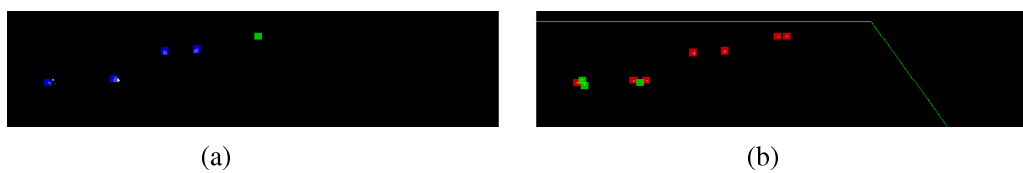


Figura 4.7: Risultati dell'elaborazione dell'immagine in Fig. 4.6: (a) Immagine binarizzata con soglia alta, (b) Immagine differenza.

4.3 Prestazioni dell'individuazione di un faro

Le prestazioni che verranno descritte riguardano il risultato delle fasi di preelaborazione e di riconoscimento fari. Nelle tabelle seguenti sono riportati le distanze espresse in metri alle quali si individua un faro. Questi dati sono stati ricavati dall'analisi delle prove statiche fornite dal C.R.F.

Tipo di immagine	Tipo di fari	Distanza 150	Distanza 100	Distanza 50
binarizzazione alta	anteriori	?	SI	SI
binarizzazione alta	posteriori	NO	NO	NO
differenza	anteriori	?	SI	NO
differenza	posteriori	NO	SI	SI

Tabella 4.1: Prestazioni sulla distanza di individuazione con abbaglianti spenti.

Purtroppo, non sono state fornite prove statiche con abbaglianti accesi. Dall'analisi dei filmati, è possibile riportare i risultati nella Tabella 4.2.

Tipo di immagine	Tipo di fari	Distanza 150	Distanza 100	Distanza 50
binarizzazione alta	anteriori	?	SI	SI
binarizzazione alta	posteriori	NO	NO	NO
differenza	anteriori	?	SI	NO
differenza	posteriori	NO	NO	SI

Tabella 4.2: Prestazioni sulla distanza di individuazione con abbaglianti accesi.

I punti interrogativi indicano che, con le immagini a disposizione e una calibrazione della camera approssimativa, anche qualitativamente non si è sicuri del risultato. Inoltre, nelle Tabelle 4.1 e 4.2 non vengono specificati i casi in cui sono accesi i fari di stop o gli abbaglianti dei veicoli incrociati. Entrambi i casi migliorano notevolmente le prestazioni corrispondenti. Un veicolo incrociato avente fari abbaglianti accesi viene sicuramente individuato a 150 metri e un veicolo che sta frenando è ben identificabile anche a 100 metri di distanza.

Come specificato nel Capitolo 2, l'immagine binarizzata con soglia alta fornita dalla preelaborazione non contiene mai illuminazione esterna. Questo significa che se viene riconosciuto un faro in questa immagine, questi è sicuramente un veicolo e l'algoritmo può comandare immediatamente lo spegnimento dei fari abbaglianti se questi sono accesi. Dai risultati nelle Tabelle 4.1 e 4.2 si ottiene che è sufficiente l'elaborazione di una singola immagine per spegnere gli abbaglianti se si incrocia un veicolo a 100 metri oppure a 150 se ha i fari abbaglianti accesi. Ulteriori considerazioni si rimandano al Capitolo 6.

Il problema aperto di questo progetto è il riconoscimento di un veicolo nello stesso senso di marcia troppo in ritardo, rischiando così un abbagliamento posteriore. Nel Conclusioni di questa tesi si spiegheranno quali sono le metodologie per risolvere tale problema.

Capitolo 5

Tracking dei fari

Il primo capitolo di questa tesi ha evidenziato come il risultato del riconoscimento di un veicolo dato dall'elaborazione di una singola immagine non sia affidabile. Nonostante l'efficienza degli algoritmi di preelaborazione, la forma, l'intensità e la posizione di un faro non sono sufficienti a distinguerlo da un paletto catarifrangente a lato della carreggiata. Nel rispettare i vincoli sul riconoscimento di motoveicoli inoltre, non è possibile nemmeno discriminare un faro a seconda che ne sia presente uno a lato con la stessa morfologia [7]. E' pertanto necessario cercare di studiare il movimento del faro in una serie di immagini [9]. Lo studio completo del movimento di un oggetto, anche se solo nelle due dimensioni, comporta un lungo lavoro che esula dalle principali finalità di questo progetto. L'algoritmo prodotto si occupa di distinguere i diversi fari ed etichettarli cercando di capire per quanti *frame* un faro è presente nella sequenza: chiameremo questo valore *età*. Come spiega in maniera approfondita il Paragrafo 5.2, l'*età* permette di distinguere un veicolo da un altro oggetto luminoso perchè un faro, a differenza di altra illuminazione, non solo permane nell'inquadratura per molto più tempo ma viene comunque sempre rilevato se presente. È vero che nelle immagini elaborate sono presenti riflessi di altri oggetti ma la loro localizzazione avviene in maniera sporadica ed imprecisa a causa dell'efficienza della preelaborazione.

5.1 Algoritmo di Tracking

L'algoritmo di *tracking* si occupa di riconoscere un determinato faro per più immagini consecutivamente. Pertanto deve non solo etichettare il faro ma anche dire se il faro che individuerà nell'immagine successiva è lo stesso riconosciuto precedentemente oppure no. Come rappresentato in Figura 5.1 l'algoritmo si scompone in due moduli principali: il predittore e l'etichettatore.

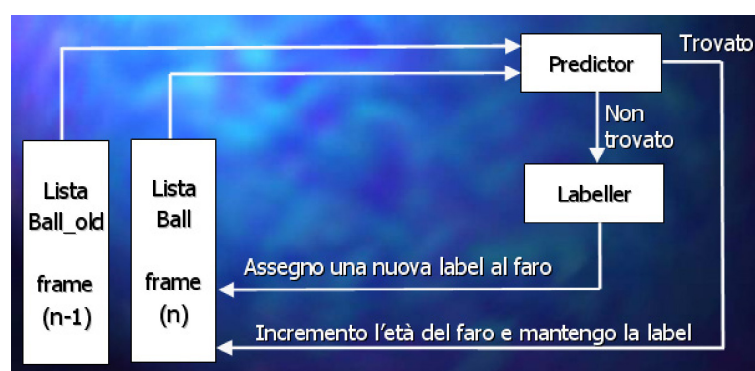


Figura 5.1: Diagramma di flusso dell'algoritmo di *tracking*.

L'algoritmo di *tracking* elabora solo i fari ottenuti dalla localizzazione nell'immagine differenza. Questo perchè ogni faro localizzato nell'immagine binarizzata con soglia alta appartiene sicuramente ad un veicolo e non ha bisogno di ulteriori elaborazioni per confermare tale risultato. In ingresso il predittore processa due liste di dati contenenti le informazioni dei fari localizzati nel frame precedente e in quello attuale. Le liste mantengono le informazioni sulla posizione, dimensione, età, etichetta, valore di confidenza e regione nella quale è più probabile localizzare un faro nel *frame* successivo.

5.1.1 Predittore

Il compito del predittore è quello di predire dove l'oggetto precedentemente riconosciuto si troverà nell'immagine successiva. È intuitivo dedurre che il faro nella successiva immagine si troverà all'incirca nella stessa posizione precedente. Ciò è vero solo se si percorre un rettilineo ed il veicolo considerato si trova ad una distanza con-

siderevole oppure se si viaggia con velocità uguale e si percorre la stessa direzione del veicolo. In tutti gli altri casi la zona possibile varia non solo a seconda della distanza del veicolo incrociato ma anche della sua velocità. Ad esempio percorrendo una curva, il faro selezionato si sposta notevolmente lungo tutta la maggior parte della larghezza dell'immagine acquisita. Nelle prime sperimentazioni di questo algoritmo la zona di ricerca veniva posta sufficientemente ampia da contenere un qualsiasi spostamento di un faro. Tuttavia, questo metodo, se pur efficace, generava numerosi errori. Essendo la zona di ricerca molto ampia poteva contenere anche un faro diverso da quello precedente, magari appartenente ad un altro veicolo. Per tali motivi, la determinazione della zona di ricerca del predittore viene calcolata dinamicamente in base alla variazione della precedente posizione del faro. Facendo l'ipotesi di individuare il faro di un veicolo lontano e che quindi varia pochissimo la sua posizione nel *frame* successivo, la zona di ricerca viene impostata ad un rettangolo di dimensioni ridotte. Le dimensioni dei suoi lati variano proporzionalmente alla variazione dello scostamento orizzontale oppure verticale della sua posizione. Si utilizza un rettangolo e non un quadrato perchè la variazione orizzontale è sempre maggiore o almeno uguale, a quella verticale.

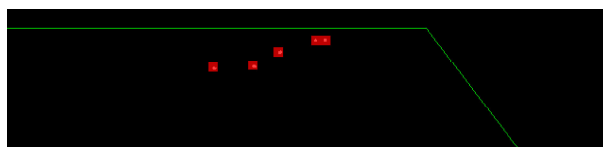


Figura 5.2: Immagine differenza con localizzazione dei fari.

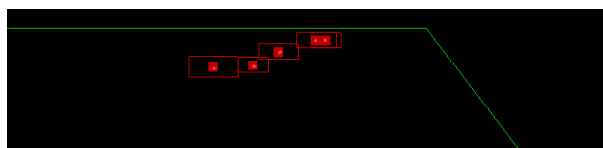


Figura 5.3: Il quadrato attorno ad un faro circoscrive la zona nella quale al passo successivo l'etichettatore andrà a cercarlo.

In Figura 5.3 viene raffigurato con un rettangolo di colore uguale a quello assegnato al faro individuato la zona di ricerca calcolata dal predittore nella quale l'etichettatore al passo successivo andrà a cercare quel determinato faro.

5.1.2 Etichettatore

L'etichettatore ha il compito di controllare la presenza di un faro nella regione individuata dal predittore: se lo trova incrementa l'età del faro, cioè il numero di immagini consecutive nelle quali è stato localizzato. L'etichettatore deve essere anche in grado di riconoscere quali sono i fari nuovi mai comparsi prima. Una volta terminata la ricerca nelle zone dettate dal predittore, se esistono altri fari l'etichettatore assegna una nuova etichetta al faro e impone l'età a zero. Al passo successivo, se questo faro sarà presente nella regione predetta, la sua età sarà incrementata di una unità. È utile fare qualche considerazione sull'andamento del moto dei fari nel tempo. Una possibilità che è stata valutata è quella di considerare come varia la dimensione del faro trovato in relazione alla sua età. Infatti quando viene incrociato un veicolo, la dimensione dell'intensità di un faro acquisita non può far altro che aumentare o al più rimanere costante. Tuttavia, se consideriamo la situazione in cui un veicolo che precede la macchina sperimentale prima frena bruscamente e poi accelera raggiungendo una velocità superiore quella del veicolo sperimentale, il fascio acquisito risulta dapprima aumentare velocemente e successivamente diminuire. In sostanza non è possibile fare qualche considerazione attendibile sull'evolversi della dimensione del faro "inseguito". Sarebbe invece interessante studiare lo spostamento del faro nel tempo. Intuitivamente, essendo il moto di un generico faro diverso dal moto di un altro oggetto potrebbe essere ragionevole studiare come si evolve il vettore spostamento tra una immagine e l'altra. Questa tecnica è esaminata nel Paragrafo 7.2 come sviluppo futuro.

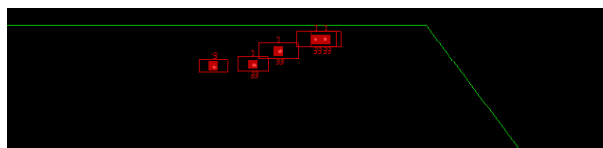


Figura 5.4: Il numero sopra il quadrato rappresenta l'etichetta del faro, mentre quello sotto visualizza l'età del faro.

5.2 Valutazione delle prestazioni dell'algoritmo di tracking

L'algoritmo di *tracking* risulta efficiente e sufficientemente veloce ai fini del progetto. Tuttavia è doveroso sottolineare alcuni errori che questo algoritmo genera.

Come si nota in Figura 5.4 sia fari dello stesso veicolo che di veicoli diversi possono avere la stessa etichetta. Questo accade quando si localizzano fari a distanze elevate di veicoli consecutivi. Dato che i veicoli sono lontani, le posizioni dei fari sono molto vicine e l'etichettatore assegna la stessa etichetta a più fari. Anche se concettualmente errato, non costituisce un problema per l'algoritmo in quanto viene incrementato al massimo l'età di un vero faro migliorando le prestazioni di localizzazione senza commettere errori. Inoltre, la stessa etichetta e la stessa età di fari di uno stesso veicolo è un vantaggio nell'individuazione di un veicolo.

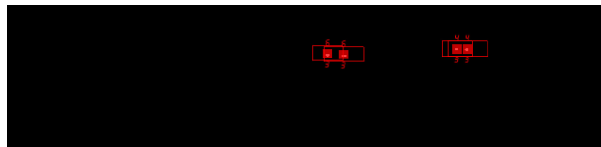


Figura 5.5: Entrambi i fari di un veicolo hanno la stessa etichetta.

Esistono invece condizioni per cui l'algoritmo di *tracking* non funziona correttamente. Ad esempio, se si percorre un percorso rettilineo con la presenza a lato di paletti catarifrangenti, esiste una combinazione fra velocità del veicolo e velocità di acquisizione tale per cui la riflessione dei paletti compare sempre nel medesimo punto nell'immagine acquisita anche se prodotta da paletti diversi. Anche se tale situazione si verifica di rado, si è reso necessario irrobustire la fase di preelaborazione con abbaglianti accesi, dato che è in questa fase che le riflessioni sono più intense. Questo ha lo scopo di eliminare la maggior quantità possibile di rumore generato dai paletti catarifrangenti. Tale variazione, però, causa una perdita di prestazioni sulla distanza di localizzazione con fari abbaglianti accesi di un veicolo percorrente la stessa direzione. Si rimanda al Capitolo 6 per ulteriori considerazioni.

Le immagini in Figura 5.6 e 5.7 mostrano come in generale distinti paletti corrispondano ad etichette con età diverse.



Figura 5.6: Immagine acquisita dalla camera.

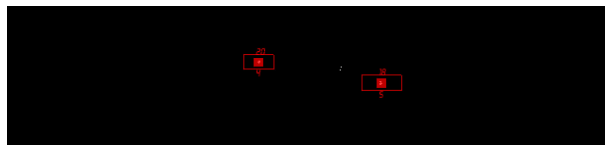


Figura 5.7: Anche se l'etichettatore etichetta con lo stesso valore due fari vicini, nel caso dei paletti catarifrangenti, questo inconveniente si verifica comunque raramente.

Le sequenze forniteci dal Centro Ricerche Fiat sono state acquisite con un *frame rate* pari a 4, il che equivale a dire che un *frame* viene acquisito ogni 250 ms. Se ad esempio il veicolo sperimentale ed un veicolo incrociato viaggiano entrambi ad una velocità pari ad 90 Km/h, il faro del veicolo incrociato viaggia ad una velocità relativa alla camera di 180 Km/h percorrendo 12,5 metri in 250 ms. Questo comporta uno notevole spostamento del faro nell'immagine acquisita dalla camera tra un *frame* e l'altro proporzionalmente alla distanza relativa del veicolo. Inoltre, anche quando si effettua un cambiamento di corsia o si percorre una curva il faro individuato varia di molto la sua posizione nell'immagine.

Un *frame rate* più elevato permetterebbe sicuramente di avere una etichettatura migliore dei fari. Infatti, acquisendo più immagini al secondo, la scena inquadrata

dalla camera varia in maniera molto inferiore tra un *frame* e l'altro. In questo modo, il predittore può restringere il campo di ricerca di un faro evitando così di etichettare erroneamente fari vicini. L'algoritmo è in grado di processare un numero maggiore di immagini al secondo. Ulteriori considerazioni si trovano nella parte dedicata agli sviluppi futuri del Capitolo 7.

5.3 Determinazione della presenza di un veicolo

Come specificato nel paragrafo precedente un faro viene distinto da un altro oggetto a seconda della sua età. Dopo numerose sperimentazioni, un faro appartiene ad un veicolo e non ad altri oggetti se ha una età maggiore di nove. Questo significa che se il faro viene inseguito per almeno dieci immagini consecutive questi identifica un veicolo. Nel caso in cui l'algoritmo conosca la posizione della carreggiata, invece, l'età deve essere maggiore di quattro per garantire la presenza di un veicolo. Questo perchè la presenza di rumore nell'immagine è molto inferiore che nel caso precedente. Se tali situazioni non si verificano, l'algoritmo non riconosce nessun veicolo e pertanto impone l'accensione dei fari abbaglianti se questi erano spenti.

È necessario precisare che tali vincoli sono stati determinati considerando come prioritario evitare di commettere errori di individuazione. Tali valori, che potrebbero essere considerati eccessivamente alti, garantiscono una certezza quasi assoluta nel riconoscimento del veicolo.

In realtà, nel caso in cui sia certa l'assenza di veicoli, l'accensione degli abbaglianti viene ritardata di 1,5 secondi per evitare un fastidioso sfarfallio dei fari nel caso sopraggiungano veicoli in successione. Lo spegnimento degli abbaglianti, invece, è immediato.

L'ultima considerazione riguarda la sensibilità agli errori dell'algoritmo di *tracking* e alla conseguente tecnica di rilevazione di un veicolo. Risulta evidente che se l'algoritmo non riconosce un faro anche per un solo frame, perde tutte le sue informazioni fino a quel momento compresa la sua età causando una inaccettabile perdita di prestazioni sulla distanza di rivelazione. Tale inconveniente si può verificare se un faro non viene riconosciuto dall'algoritmo di *Pattern Matching*, oppure se l'etichettatore

non trova il faro nella zona calcolata dal predittore. Questi sono gli unici due errori che possono causare la perdita dell'età del faro e la mancata rilevazione di un veicolo. Nonostante la notevole sensibilità agli errori dell'algoritmo di *tracking*, sia l'algoritmo di *Pattern Matching* che quello del predittore insieme all'etichettatore sono stati implementati mantenendo ampi margini di sicurezza sulla rilevazione. Questo permette di scongiurare errori critici del *tracking* ed ottenere risultati estremamente affidabili.

Capitolo 6

Risultati sperimentali

In questo capitolo vengono mostrati i risultati dei vari test effettuati, i quali hanno dimostrato le potenzialità dell' algoritmo sviluppato; si pone però l' attenzione anche sui problemi che si sono presentati. I dati di seguito esposti, ricavati dalle elaborazioni di tutte le sequenze in nostro possesso, saranno fondamentali nel guidare il Centro Ricerche Fiat alla valutazione dell' applicabilità commerciale del progetto nel breve periodo.

6.1 Tempi di esecuzione

L' algoritmo è stato testato utilizzando un calcolatore le cui caratteristiche sono indicate in Tabella 6.1. La compilazione del codice è stata ottenuta usando le opzioni di ottimizzazione adeguate per sfruttare al massimo le risorse del calcolatore.

Processore	Pentium IV
Frequenza di clock	2.80 GHz
Cache	512 KB
RAM	1035304 KB

Tabella 6.1: Caratteristiche tecniche del calcolatore utilizzato per i test.

Ovviamente il tempo di elaborazione dipende strettamente dalla tipologia delle

immagini in ingresso: la presenza di molti veicoli o di un ambiente stradale molto illuminato, costringe il programma ad eseguire un numero maggiore di operazioni.

Per valutare la velocità di esecuzione sono state fatte diverse prove analizzando sequenze video differenti e valutando i tempi medi di elaborazione. Le sequenze utilizzate sono quelle forniteci dal Centro Ricerche Fiat direttamente acquisite dalla camera Neuricam NC5000 installata sul veicolo Lancia K. In Tabella 5.2 sono elencate le sequenze processate e la loro tipologia.

Sequenza	Tipologia	N. Frame	T. acquisiz.(min)
RedFilter_M1	notturna, acquisita con abb. spenti	2516	17
RedFilter_M2	notturna, acquisita con abb. spenti	2650	18
RedFilter_M3	notturna, acquisita con abb. spenti	7315	49
RedFilter_S4	notturna, acquisita con abb. accesi	1086	7
RedFilter_S5	notturna, acquisita con abb. accesi	3328	22
RedFilter_S6	notturna, acquisita con abb. accesi	3754	25
RedFilter_S7	notturna, acquisita con abb. accesi	7259	48

Tabella 6.2: Sequenze analizzate durante i test.

Come menzionato nel primo Capitolo di questa tesi, la necessità dell'elaborazione in tempo reale ha imposto vincoli precisi sui tempi di esecuzione dell'algoritmo. Le seguenti tabelle riportano i tempi di elaborazione divisi per tipologia di sequenza.

Sequenza	t. min (ms)	t. max (ms)	t. medio (ms)
RedFilter_M1	60	296	119
RedFilter_M2	57	302	120
RedFilter_M3	58	312	117
Media Tot.	58,33	303,33	118,66

Tabella 6.3: Tempi d'elaborazione di un *frame* per ciascuna sequenza.

Le Tabella 6.3 e 6.4 mostrano come i tempi di esecuzioni medi siano abbondantemente sotto il limite massimo imposto dalla sperimentazione in tempo reale. Questo limite attualmente vale 250 ms. Tuttavia, l'algoritmo impiega mediamente più tempo

Sequenza	t. min (<i>ms</i>)	t. max (<i>ms</i>)	t. medio (<i>ms</i>)
RedFilter_S4	73	228	117
RedFilter_S5	65	340	143
RedFilter_S6	72	405	154
RedFilter_S6	62	477	143
Media Tot.	68	362,5	139,25

Tabella 6.4: Tempi d'elaborazione di un *frame* per ciascuna sequenza.

nel processare le immagini acquisite con fari abbaglianti accesi. Questo è dovuto al maggior carico computazionale della parte di preelaborazione.

Comunque, i tempi di elaborazione dipendono dal calcolatore utilizzato: un calcolatore più potente diminuirebbe notevolmente i tempi di esecuzione dell'algoritmo.

6.2 Prestazioni quantitative

L'algoritmo che è stato sviluppato per questo lavoro di tesi è completo e non necessita di ulteriori moduli per ottenere i risultati richiesti. Ovviamente, può essere migliorato rendendolo più veloce e affidabile ma già in questa sua fase di sviluppo è in grado di fornire risultati soddisfacenti.

Le Tabelle 6.5 e 6.6 riportano il numero di errori nei risultati dell'algoritmo. Il risultato dell'algoritmo è rappresentato da un solo bit: se vale zero i fari abbaglianti vengono spenti o restano tali; se vale uno, i fari abbaglianti vengono accesi se non lo sono già. Dato che non è nostro compito cablare un sistema fisico di controllo dei fari, il programma visualizza l'immagine acquisita correlata di una icona posizionata in alto a destra, rappresentante lo stato dei fari pilotato dall'algoritmo.

La Tabella 6.5 riporta i risultati dell'elaborazione delle immagini acquisite con fari abbaglianti spenti. I risultati sono eccellenti. Osservando i valori medi se ne deduce che percorrendo per circa mezzora una qualsiasi strada ed incrociando una novantina di veicoli, l'algoritmo riconosce correttamente tutti i veicoli non scambiando mai illuminazione stradale per automobili ma abbagliando erroneamente per meno di



Figura 6.1: Icona fari anabbaglianti accesi.



Figura 6.2: Icona fari abbaglianti accesi.

un secondo uno solo di tutti i veicoli incrociati. Per falsi positivi, vedi Figura 6.3, si intendendo quegli oggetti riconosciuti come un veicolo ma che non lo sono nella realtà. Viceversa, i falsi negativi sono veicoli effettivamente presente nell'inquadratura ma non riconosciuti dall' algoritmo. I valori dei falsi positivi e negativi indicano errori compiuti dall' algoritmo anche solo per un semplice *frame*. Gli errori di questo genere causano o l'abbagliamento anche se istantaneo di un veicolo oppure lo spegnimento degli abbaglianti in assenza di veicoli inquadrati.

Sequenza	Durata seq. (min)	Numero veicoli	Falsi Positivi	Falsi Negativi
RedFilter_M1	17	110	0	2
RedFilter_M2	18	80	0	0
RedFilter_M3	49	75	0	0
Media Tot.	28	88	0	1

Tabella 6.5: Errori nell'elaborazione con fari abbaglianti spenti.



Figura 6.3: Esempio di falso positivo. L'algoritmo comanda lo spegnimento degli abbaglianti (vedi icona) nonostante non sia presente nessun veicolo.

Dalla Figura 6.4 alla Figura 6.13 vengono visualizzati scenari particolari in cui l'algoritmo ha controllato i fari in maniera corretta durante l'elaborazione a fari abbaglianti spenti.



Figura 6.4: Caso tipico di corretta rilevazione di una serie di veicoli.



Figura 6.5: L'algoritmo decide correttamente di non spegnere i fari abbaglianti nonostante sia presente molto rumore.



Figura 6.6: Riconoscimento di un veicolo a notevole distanza.

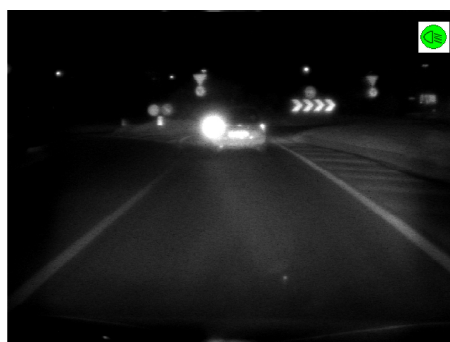


Figura 6.7: Riconoscimento di un veicolo avente un faro posteriore non funzionante.

Interessante è notare come nonostante il percorso stradale sia svariatamente complesso, l'algoritmo fornisca dei risultati corretti. Percorsi che includono gallerie, segnaletica orizzontale voluminosa, luci di pericolo lampeggianti ed incroci semaforici non costituiscono elementi di errore. Nemmeno, come visualizzato in Figura 6.7, un fanale posteriore danneggiato evita il riconoscimento di un veicolo. Anche i motoveicoli sono riconosciuti come descritto nelle specifiche di progetto.

Non solo i risultati sono attendibili ma spesso sono ottenuti in tempi inferiori di



Figura 6.8: Corretto controllo dei fari in caso di sorpasso.



Figura 6.9: Corretto riconoscimento di un motociclo.



Figura 6.10: Corretto controllo dei fari in caso di sorpasso.



Figura 6.11: Riconoscimento di un veicolo a lato in curva.

quello che un controllo umano potrebbe ottenere. Si prendano ad esempio le Figure 6.12 e 6.13. Nel primo caso, il veicolo incrociato appare improvvisamente trovandosi dal lato opposto di un cavalcavia. L'algoritmo impiega solamente l'elaborazione di un singolo *frame* per riconoscerlo; il che vale a dire secondo la Tabella 6.3, mediamente 118 ms: tempo nettamente inferiore a quello impiegato da un conducente per controllare manualmente i fari. Anche in Figura 6.13 si presenta un caso analogo. Un veicolo incrociato appare improvvisamente sul lato opposto della corsia percorrendo una curva. Come nel caso precedente, l'algoritmo impiega circa 118 ms per spegnere i fari abbaglianti. Tali prestazioni confermano non solo una buona analisi e progettazione dell'algoritmo ma risaltano le potenzialità delle tecniche di visione artificiale applicate al controllo automatico di veicoli.

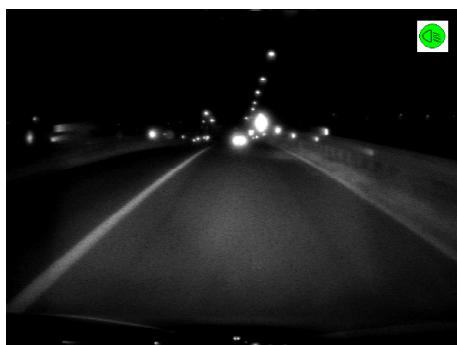


Figura 6.12: Riconoscimento di un veicolo comparso all'improvviso percorrendo un cavalcavia.

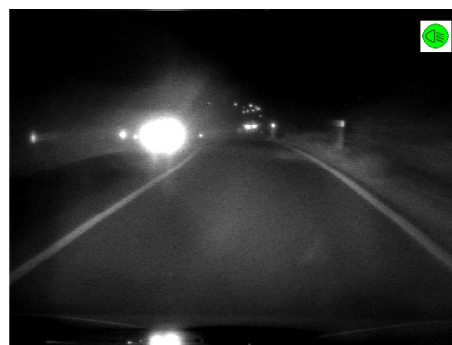


Figura 6.13: Riconoscimento di un veicolo comparso all'improvviso all'approssimarsi di una curva.

La Tabella 6.6, invece, riporta i risultati dell'elaborazione delle immagini acquisite con fari abbaglianti accesi. Come abbiamo già accennato, l'elaborazione in questa situazione è maggiormente soggetta ad errori a motivo della forte riflessione degli oggetti a lato della carreggiata.

Sequenza	Durata seq. (<i>min</i>)	Numero veicoli	Falsi Positivi	Falsi Negativi
RedFilter_S4	7	12	1	0
RedFilter_S5	22	29	8	0
RedFilter_S6	25	19	2	1
RedFilter_S7	48	76	11	2
Media Tot.	26	34	6	1

Tabella 6.6: Errori nell'elaborazione con fari abbaglianti spenti.

Dalla Tabella 6.6 traspare, che anche nel caso di elaborazione con abbaglianti accesi, i risultati sono più che soddisfacenti. Considerando la varietà degli ambienti stradali e la scrupolosa precisione nel misurare gli errori, si nota come in 26 minuti di tragitto mediamente soltanto uno dei 26 veicoli incrociati venga solo per qualche istante abbagliato. Come era prevedibile, però, il numero dei falsi positivi è aumentato causando lo spegnimento dei fari senza nessun veicolo incrociato.

Dalle Figure 6.16 alla Figura 6.21 vengono visualizzati scenari particolari in cui



Figura 6.14: Caso di falso positivo: non è presente nessun veicolo.



Figura 6.15: Caso di falso positivo: l'algoritmo riconosce come veicoli degli insoliti fari lampeggianti presenti in galleria.

l'algoritmo ha controllato i fari in maniera corretta durante l'elaborazione a fari abbaglianti accesi.



Figura 6.16: Corretta rilevazione di un veicolo a distanza considerevole.



Figura 6.17: L'algoritmo decide correttamente di non accendere i fari abbaglianti nonostante sia presente molto rumore.

Le stesse considerazioni fatte per i risultati ottenuti con abbaglianti spenti valgono anche per quelli accesi. Rilevante è osservare in Figura 6.18 come l'algoritmo riconosca tutti i veicoli compresi quelli di dimensioni molto maggiori come i camion. Lo svantaggio rimane comunque quello di riconoscere qualche volta veicoli quando effettivamente non sono presenti. A questo proposito, vale la pena ricordare che in un percorso medio, i fari abbaglianti rimangono accesi per meno tempo che quelli anabbaglianti e che comunque il danno maggiore di questo svantaggio è quello di



Figura 6.18: Riconoscimento di diversi veicoli contemporaneamente.



Figura 6.19: Corretto riconoscimento in situazione di sorpasso.



Figura 6.20: Riconoscimento di un veicolo notevolmente distante.



Figura 6.21: Corretto riconoscimento di un veicolo in curva.

spegnere gli abbaglianti per due o tre secondi al massimo e non quello di abbagliare pericolosamente un veicolo in prossimità.

Nel Capitolo 7 verranno prese in esame alcune considerazioni opportune al fine di migliorare le prestazioni con fari abbaglianti accesi.

6.3 Prestazioni sulla distanza di localizzazione

La correttezza dei risultati è fondamentale per giudicare in modo positivo il funzionamento del programma. Nel progetto a noi commissionato, però, per confermare che il funzionamento sia corretto, non solo è necessario garantire l'elaborazione in tempo reale e riconoscere tutti i veicoli come tali scartando altri oggetti come rumore.

Per ottenere un controllo intelligente dei fari è necessario agire sul loro stato evitando di abbagliare veicoli incrociati. In altre parole è necessario spegnere i fari abbaglianti quando un veicolo è sufficientemente distante onde garantire che non venga abbagliato anche se per pochi istanti.

Non disponendo di dati di calibrazione precisi e utilizzando un sistema di visione monoculare è impossibile riuscire a calcolare con sufficiente precisione la distanza di un veicolo inquadrato dalla camera. Tale svantaggio permette solamente di stimare la distanza di un veicolo incrociato.

Anche in questo caso è necessario distinguere i risultati per elaborazione di immagini acquisite con fari abbaglianti spenti o accesi.

Stesso senso di marcia (m)	Senso di marcia opposto (m)
> 80	> 120

Tabella 6.7: Distanza di riconoscimento di un veicolo con fari abbaglianti spenti.

Stesso senso di marcia (m)	Senso di marcia opposto (m)
> 50	> 100

Tabella 6.8: Distanza di riconoscimento di un veicolo con fari abbaglianti accesi.

I valori indicati nelle tabelle si riferiscono non all'istante in cui un faro viene individuato per la prima volta ma bensì all'istante in cui il faro viene riconosciuto come appartenente ad un veicolo. Essendo dati approssimativi viene indicato il peggior valore osservato nella distanza di riconoscimento. Inoltre le tabelle non considerano le prestazioni ottenute incrociando veicoli che procedono con abbaglianti accesi o che stanno frenando. In tali condizioni il valore minimo della distanza di riconoscimento è in ogni caso superiore ai 120 metri.

La distanza minima osservata nel riconoscimento di un veicolo percorrente lo stesso senso di marcia contenuto nella Tabella 6.8, potrebbe risultare critico nel caso in cui il veicolo sperimentale, viaggiando con fari abbaglianti accesi, raggiunga un altro vei-

colo che procede nel medesimo senso di marcia: secondo le stime nel caso peggiore, fino a quando i veicoli non si trovano a distanza di 50 metri, il controllo dei fari non ordina lo spegnimento causando un abbagliamento posteriore del veicolo che precede. Questo potrebbe risultare inaccettabile per una applicazione commerciale del progetto. Resta comunque il fatto che si sta parlando di stime considerando il caso peggiore e quindi non forniscono un metro di valutazione completamente attendibile.

Ad ogni modo, questo rappresenta il principale problema dell'algoritmo. Fortunatamente i casi in cui si verifica questo inconveniente sono rari essendoci diverse situazioni che scongiurano questo problema. In primo luogo il problema si verifica solo quando gli abbaglianti sono accesi. Ancora, se il veicolo inquadrato frena anche per un quarto di secondo, l'algoritmo identifica immediatamente il veicolo anche a distanze superiori ai 100 metri. Oppure se nell'avvicinamento con abbaglianti accesi ad un veicolo A, viene incrociato un altro veicolo B nel senso di marcia opposto, l'algoritmo riconosce B e comanda lo spegnimento dei fari abbaglianti iniziando a elaborare le immagini con i relativi parametri. Siccome quest'ultima elaborazione ha maggiori prestazioni sulla distanza (vedi Tabella 6.7) l'algoritmo riconosce il veicolo A che precedentemente con abbaglianti accesi non aveva nemmeno individuato.

Si segnala comunque, che il principale problema di un abbagliamento posteriore è il fastidioso riflesso sullo specchietto retrovisore interno al veicolo. Esistono già sistemi commercializzati che risolvono questo inconveniente. Ad esempio esistono gli specchietti retrovisori interni antiabbagliamento automatici. Essendo costruiti con materiali detti cromogenici variano le proprie caratteristiche ottiche a seconda dell'intensità luminosa della luce incidente.

Da questa analisi risulta evidente che è necessario riuscire non solo ad avere dei dati affidabili sulla distanza di riconoscimento di un veicolo ma anche di provare l'algoritmo direttamente sulla macchina sperimentale controllando fisicamente lo stato dei fari. In questo modo è possibile verificare quanto sia critico il ritardo nel riconoscimento con abbaglianti accesi di un veicolo percorrente lo stesso senso di marcia.

Capitolo 7

Conclusioni

In questo lavoro di tesi è stato affrontato il problema del riconoscimento di veicoli in ambienti notturni in percorsi stradali strutturati mediante visione artificiale. È stato impiegato un sistema di acquisizione monoculare utilizzando una camera a toni di grigio con uscita a 10 bit. L'algoritmo che è stato sviluppato punta a sfruttare la differenza di luminosità tra i fari dei veicoli e il resto dell'illuminazione stradale. Per questo motivo è stato applicato un filtro rosso all'obiettivo della camera e sono state create specifiche funzioni di preelaborazione che mirano a risaltare le caratteristiche radiometriche dei fasci luminosi dei fari. Successivamente si è passati alla individuazione e alla localizzazione dei probabili fari. Dato che la riflessione della segnaletica orizzontale risulta particolarmente accentuata è stato necessario introdurre delle funzioni capaci di individuare la posizione della carreggiata all'interno dell'immagine. Sfruttando la trasformata di Hough, viene individuato un pixel che rappresenta il punto di fuga dell'immagine. Dalle coordinate di questo pixel è stato possibile ricavare la zona all'esterno della carreggiata e quindi eliminare tutti quegli oggetti che appartengono a tale zona. Dato che da una singola immagine è estremamente difficile riconoscere con sicurezza un veicolo è stato necessario studiare per quanto tempo un probabile faro localizzato è presente. Se un faro è presente per un determinato numero di immagini consecutive, questi individua un veicolo e non altri oggetti.

L'algoritmo si è dimostrato efficiente indipendentemente da quanto è complesso il percorso stradale. Il discreto numero di sequenze testate è garanzia dell'affidabi-

lità dell'algoritmo. I tempi di esecuzione sono abbondantemente inferiori del vincolo imposto dal sistema di acquisizione montato sul veicolo sperimentale. Le prestazioni sulla correttezza dei risultati sono molto migliori delle aspettative iniziali. Una considerazione importante riguarda le prestazioni dell'algoritmo in situazioni particolari. Come si è esaminato nel capitolo precedente esistono situazioni in cui il controllo manuale dei fari risulta troppo lento. Ad esempio se si percorre una curva o un cavalcavia potrebbe comparire improvvisamente nella corsia opposta un altro veicolo. L'algoritmo impiega mediamente poco meno di 140 ms per abbassare gli abbaglianti. Tale reattività non solo è estremamente affidabile ma anche sicuramente molto maggiore del tempo che un guidatore impiegherebbe ad azionare la leva dello spegnimento dei fari. Essendo tali situazioni molto comuni nella guida di notte e nello stesso tempo molto pericolose, come menzionato nell'introduzione di questa tesi, crediamo che l'affidabilità e le prestazioni dell'algoritmo possano garantire una base solida su cui implementare un sistema di controllo dei fari effettivamente commercializzabile capace di garantire maggiore sicurezza nella guida notturna.

7.1 I limiti di questa analisi

Il limite di questo programma è stato descritto nel paragrafo sulle prestazioni nella distanza di localizzazione di un veicolo del precedente capitolo.



Figura 7.1: Limite del programma: elaborando nella modalità con abbaglianti accesi non riconosce un veicolo che precede se distante meno di 50 metri.

Se il veicolo sperimentale, viaggiando con fari abbaglianti accesi, raggiunge un altro veicolo che procede nel medesimo senso di marcia, l'algoritmo non riconosce il veicolo come tale se non a distanza non maggiore di 50 metri. È vero che, come accennato precedentemente, non si hanno dati sufficientemente precisi sulla distanza di calibrazione e che un risultato così scadente è veramente difficile da ottenersi a motivo dei molteplici casi che lo scongiurano. Tuttavia, tale problema rappresenta il limite principale dell'algoritmo.

La soluzione di questo problema va ricercata in un miglioramento della fase di preelaborazione con abbaglianti accesi. La modifica delle soglie di binarizzazione delle immagini da preelaborare e l'aumento del numero di cicli eseguiti dal filtro passa basso nell'immagine differenza non sono sufficienti a garantire una estrapolazione dei fasci luminosi dei fari posteriori di veicoli lontani.

7.2 Sviluppi futuri

Gli sviluppi futuri di questo lavoro di tesi saranno orientati a migliorare le capacità dell'algoritmo e a risolvere i problemi che sono emersi durante il suo sviluppo. In particolare dovranno essere analizzati i seguenti aspetti.

7.2.1 Condizioni ambientali differenti

E' necessario sperimentare l'algoritmo con immagini acquisite in diverse condizioni ambientali come foschia, pioggia, crepuscolo. Anche se l'algoritmo è stato progettato per funzionare anche in queste situazioni, manca ancora l'elaborazione in tali condizioni e pertanto non è possibile calcolare alcuna prestazione.

7.2.2 Restringimento dell'inquadratura

Il restringimento del campo di acquisizione aumenterebbe le prestazioni sui tempi di esecuzione dell'algoritmo.

La parte inferiore delle immagini acquisite contengono esclusivamente il cofano del veicolo e non viene utilizzata. Inoltre, dato che la determinazione della posizione

della carreggiata viene determinata molte meno volte dallo studio dell'illuminazione stradale che dall'analisi delle linee sulla carreggiata, potrebbe essere utile ridurre l'inquadratura della camera eliminando la parte superiore e la parte inferiore dell'immagine senza modificare la posizione e la dimensione della zona d'interesse nella ricerca dei veicoli. Su specifica richiesta del Centro Ricerche Fiat, abbiamo testato l'algoritmo simulando una acquisizione non più 640x478 ma 640x320 *pixel* mantenendo la stessa calibrazione esterna della camera. Le simulazioni hanno confermato tutte le prestazioni sulla distanza di riconoscimento e sulla correttezza dei risultati. Inoltre si è osservato un netto aumento sulla velocità di esecuzione dell'algoritmo (vedi Tabelle 7.1 e 7.2).

Sequenza	t. min (<i>ms</i>)	t. max (<i>ms</i>)	t. medio (<i>ms</i>)
RedFilter_M1_cut	45	55	48
RedFilter_M2_cut	44	56	48
RedFilter_M3_cut	45	64	47
Media Tot.	44,66	58,33	47,66

Tabella 7.1: Tempi d'elaborazione di un *frame* per ciascuna sequenza.

Sequenza	t. min (<i>ms</i>)	t. max (<i>ms</i>)	t. medio (<i>ms</i>)
RedFilter_S4_cut	71	126	76
RedFilter_S5_cut	70	178	82
RedFilter_S6_cut	70	171	81
RedFilter_S7_cut	69	230	74
Media Tot.	70	176,25	78,25

Tabella 7.2: Tempi d'elaborazione di un *frame* per ciascuna sequenza.

Un tale restringimento dell'inquadratura e la relativa diminuzione dei tempi di elaborazioni supportati dalle medesime prestazioni sui risultati fanno sì che sia possibile aumentare sensibilmente il *frame rate* delle sequenze e quindi aumentare la precisione di inseguimento dell'algoritmo di "tracking".

7.2.3 Validazione delle prestazioni

Come è stato detto nel capitolo precedente, non disponendo di dati di calibrazione precisi e utilizzando una visione monoscopica, è impossibile riuscire a calcolare con sufficiente precisione la distanza di un veicolo inquadrato dalla camera.



Figura 7.2: In rosso viene indicato la distanza del veicolo incrociato.



Figura 7.3: Il radar funziona efficacemente anche con più veicoli inquadrati.

Tale svantaggio permette solamente di stimare la distanza di un veicolo incrociato. Sarebbe importante ricavare delle distanze precise ai fini di validare i risultati ottenuti. Questo è fattibile utilizzando un radar montato sul veicolo sperimentale, il quale corredato da uno specifico programma riesce a calcolare la distanza di un veicolo individuato nella scena.

Nelle Figure 7.1 e 7.2 si visualizza un esempio di funzionamento del radar con re-

lativa restituzione della distanza calcolata dell'oggetto indicato da una piccola crocetta rossa.

7.2.4 Flusso ottico

Il flusso ottico costituisce la rappresentazione delle traiettorie seguite da ogni pixel degli oggetti inquadrati durante il loro movimento rispetto alla telecamera. Questa definizione è però basata su ciò che si vede e mal si adatta ad una interpretazione rigorosa. Si definisce in realtà flusso ottico l'insieme dei vettori rappresentanti le velocità (proiettate sul piano immagine) di tutti i punti di un oggetto che si muove rispetto ad un osservatore. I vettori del flusso ottico hanno, infatti, intensità proporzionale allo spostamento tra due *frame* consecutivi e quindi rappresentano delle vere e proprie velocità.

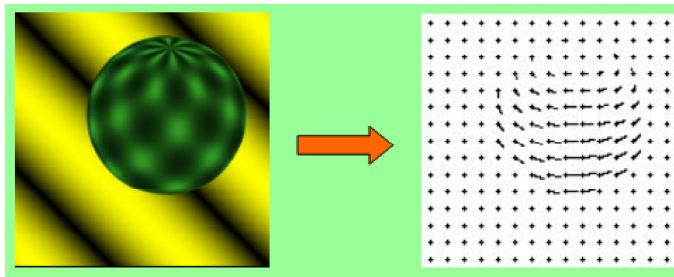


Figura 7.4: Rappresentazione del flusso ottico ricavato dal movimento della sfera.

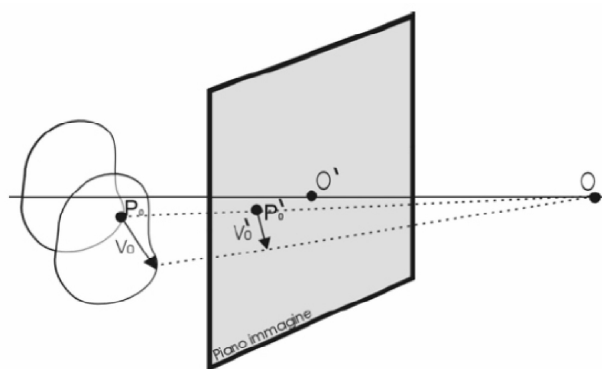


Figura 7.5: Rappresentazione dello spostamento di un oggetto.

Si consideri quindi un oggetto rigido in moto rispetto ad una telecamera O : indicando con P_0 e P'_0 un punto dell'oggetto e il corrispondente punto sul piano immagine ad un certo istante, se P_0 possiede velocità V_0 relativa alla camera, al suo spostamento $V_0 \cdot \delta t$ nell'intervallo di tempo δt corrisponderà un analogo spostamento del punto P'_0 sul piano immagine di valore $V'_0 \cdot \delta t$ dove V_0 e V'_0 sono la derivata dei vettori che congiungono il centro ottico O del sistema con i punti P_0 e P'_0 , ossia gli spostamenti cui questi due punti sono soggetti nell'intervallo di tempo δt . Usando il legame proiettivo tra i due segmenti $\overline{OP_0}$ e $\overline{OP'_0}$ risulta assegnato un vettore spostamento ad ogni punto P_i dell'immagine. L'insieme di questi vettori costituisce il campo di spostamenti che, considerati nell'unità di tempo δt piccola a piacere (limitata inferiormente dal tempo di acquisizione di due *frame* consecutivi), permettono di ricavare un campo di velocità. Il flusso ottico è definito come il campo vettoriale di tali velocità.

Dato che il movimento dei fari è diverso da quello di altri oggetti nell'immagine, l'analisi del flusso ottico potrebbe essere in grado di eliminare ulteriormente rumore dall'immagine e quindi ottenere prestazioni di distanza sulla localizzazione molto maggiori.

Appendice A

Specifiche della telecamera utilizzata

A.1 Descrizione generale

La NC5000 CLinkCam è una camera digitale industriale compatta con una interfaccia ad alta velocità CameraLink. Si connette direttamente con lo standard CameraLink *frame grabber*.

A.2 Descrizione tecnica

La NC5000 monta un sensore ottico CMOS con risoluzione 640x480 pixel (VGA) a toni di grigio. Ha un *range* dinamico di 120 dB con risposta logaritmica. Produce 10 bit in uscita per una intensità luminosa incidente che varia da 1mW/sq.m a 1000 W/sq.m. La velocità di acquisizione è di 8 Mpixel/s il che significa 24 fps a massima



Figura A.1: NC5000 CLinkCam

risoluzione VGA. La camera è dotata di 3 ingressi e uscite ottiche e la finestra di acquisizione è programmabile. L'obiettivo ha un diametro di 8 mm.

Nel progetto Incrocio veicoli la camera è stata utilizzata a piena risoluzione acquisendo quattro *frame* al secondo.

Appendice B

Manuale di utilizzo del software

Il programma è stato scritto in linguaggio *C* e sviluppato all'interno di GOLD¹[4], un sistema software d'assistenza alla guida automatica di autoveicoli, realizzato presso il Laboratorio di Visione Artificiale dell'Università degli Studi di Parma. Il modulo *AHC (Automatic HeadLights Control)* è stato realizzato per questo lavoro e contiene tutte le funzionalità del progetto implementato. Per poter utilizzare questo modulo è necessario modificare in maniera adeguata il *file gold.ini* utilizzato per la configurazione generale di *GOLD*. All'interno di questo *file* si trovano le impostazioni dei parametri che il programma legge all'avvio e in base al quale stabilisce quali funzionalità rendere accessibili all'utente. In questo caso occorre modificare il file in modo che sia presente la riga

AHC = y

Una volta attivato questo modulo, l'interfaccia grafica apparirà come in Figura B.1.

Utilizzando il pulsante *Panel* è possibile accedere a svariate finestre che permettono di modificare importanti parametri dell'elaborazione per cercare di ottenere prestazioni migliori al variare della tipologia delle sequenze di immagini processate.

¹*Generic Obstacle and Lane Detection*

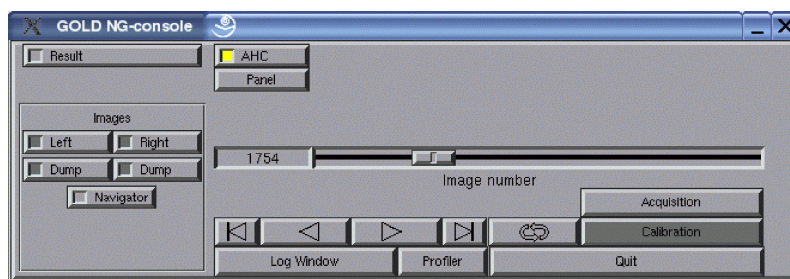


Figura B.1: Interfaccia grafica di Gold.

AHC MainPanel

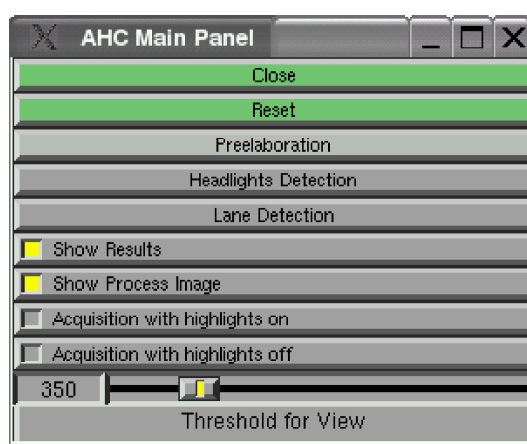


Figura B.2: Pannello principale.

Il pannello di comando principale del modulo AHC è mostrato in Figura B.2 e le sue funzioni sono:

Preelaboration: Permette di modificare i parametri della fase di preelaborazione.

Headlights Detection: Consente di aumentare o diminuire i valori di correlazione dell'algoritmo di *Pattern Matching*.

Lane Detection: Permette di accedere al pannello di configurazione della funzione di riconoscimento della posizione della carreggiata.

Show Results: Permette di visualizzare i risultati delle elaborazioni dell'algoritmo. È

necessario selezionare questa opzione per visualizzare le operazioni eseguite dal programma.

Show Process Image: Visualizza l'immagine acquisita in ingresso e lo stato dei fari in uscita mediante due distinte icone.

Acquisition with Headlights ON: Dato che spesso le sequenze possono essere acquisite solo con fari abbaglianti accesi, selezionando questa opzione, vengono mantenuti tutte le impostazioni con fari abbaglianti accesi per tutta la sequenza consentendo comunque l'elaborazione delle immagini.

Acquisition with Headlights OFF: Come sopra, ma in questo caso, la sequenza è stata acquisita interamente con abbaglianti spenti.

Threshold for View: Permette di regolare la conversione della immagine visualizzata selezionando Show Results per poterne apprezzare meglio i particolari a seconda dell'illuminazione ambientale.

AHC Preelaboration SubPanel

Questo pannello permette di variare le impostazioni della fase di preelaborazione e visualizzarne i risultati.

Show Binaryzation Diference: Visualizza l'immagine differenza binarizzata.

Show Binaryzation High: Visualizza l'immagine binarizzata con soglia alta.

Show Diference Image: Visualizza l'immagine differenza.

Threshold for Delete Noise in Input Image with HL ON: Permette di modificare la soglia per la quale i pixel aventi valori radiometrici inferiori vengono posti a zero. Questa soglia è riferita all'elaborazione con abbaglianti accesi.

Threshold for Delete Noise in Input Image with HL OFF: Come sopra ma applicata all'elaborazione con fari abbaglianti spenti.

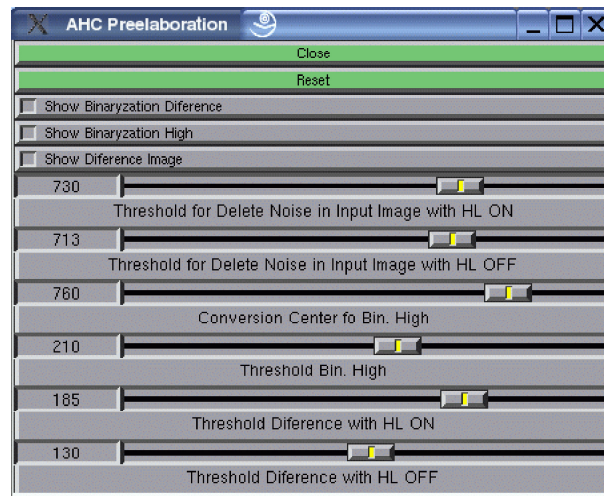


Figura B.3: Pannello preelaborazione

Conversion Center for Bin. High: Modifica il valore radiometrico del centro di conversione dell'immagine binarizzata con soglia alta.

Threshold Bin. High: Modifica la soglia di binarizzazione dell'immagine alta.

Threshold Diference with HL ON: Permette di cambiare la soglia di binarizzazione dell'immagine differenza per l'elaborazione con abbaglianti accesi.

Threshold Diference with HL OFF: Permette di cambiare la soglia di binarizzazione dell'immagine differenza per l'elaborazione con abbaglianti spenti.

AHC HeadLights Detection SubPanel

Il pannello in Figura B.4 permette di variare le impostazioni della fase di localizzazione dei fari. Attraverso gli *slider* è possibile modificare il valore massimo della differenze fra il faro da ricercare e l'*i-esimo pattern*. È possibile modificare i valori con un passo di 255 perchè è la differenza fra un pixel bianco ed un nero.

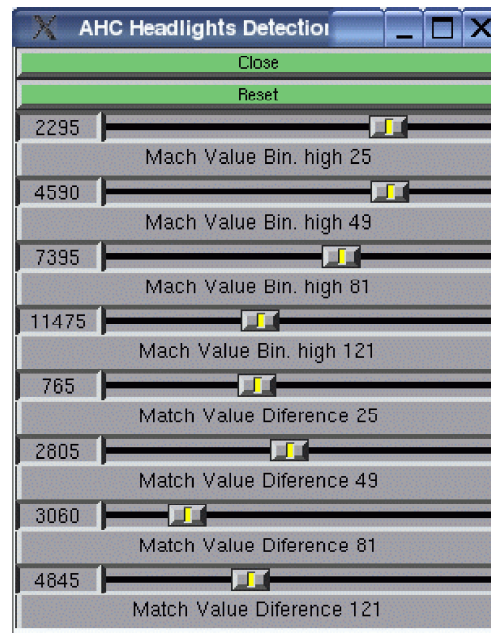


Figura B.4: Pannello localizzazione dei fari.

AHC Lane Detection SubPanel

Il pannello *AHC Lane Detection* in Figura B.5 permette di variare le impostazioni della fase di riconoscimento della carreggiata e visualizzarne i risultati. Permette inoltre di visualizzare i risultati della trasformata di Hough.

Show Edge Image Line Lane: Permette di visualizzare i contorni delle linee sulla strada.

Show Accumulator Image: Visualizza lo spazio accumulatore della trasformata di Hough.

Show Hough Results: Visualizza le linee ricavate e il punto di fuga calcolato.

Threshold for Edge Vertical Illumination: Permette di variare la soglia di binarizzazione per l'immagine contenente i contorni dell'illuminazione verticale stradale.

Threshold for Edge Lane Detection with HL ON: Permette di variare la soglia di

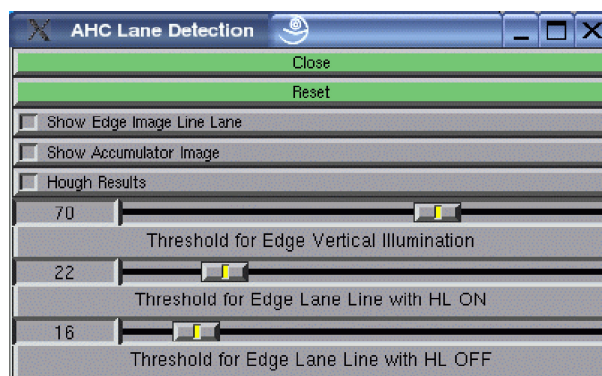


Figura B.5: Pannello di individuazione della posizione della carreggiata.

binarizzazione per l'immagine contenente i contorni delle linee sul manto stradale elaborando con fari abbaglianti accesi.

Threshold for Edge Lane Detection with HL OFF: Permette di variare la soglia di binarizzazione per l'immagine contenente i contorni delle linee sul manto stradale elaborando con fari abbaglianti spenti.

Bibliografia

- [1] C. Bellotti, F. Bellotti, A. De Gloria, L. Andreone e M. Mariani: Developing a near infrared based night vision system. *IEEE Intelligent vehicle Symposium*, 14-17 giu. 2004.
- [2] M. Kagesawa, S. Ueno, K. Ikeuchi e H. Kashiwagi: Recognizing vehicles in infrared images using IMAP parallel vision board. *IEEE Transaction on Intelligent Transportation System*, vol. 2(1):pp. 10-17, mar. 2001.
- [3] S. Nagumo, H. Hasegawa, N. Okamoto: Extraction of Forward Vehicles by Front-Mounted Camera Using Brightness Information. *IEEE CCECE Canadian Conference on Electrical and Computer Engineering*, 4-7 mag. 2003.
- [4] M. Bertozzi e A. Broggi. GOLD: A Parallel Real-Time Stereo Vision System for Generic Obstacle and Lane Detection. *IEEE Transaction on Image Processing*, vol. 7(1):pp. 62-81, gen. 1998.
- [5] M.-Y. Chern e P.C. Hou: Teh Lane Recognition and Vehicle Detection at night fo a Camera-Assisted Car on Highway. *IEEE International Conference on Robotics and Automation*, 2003.
- [6] M.S. von Trzebiatowski, A. Gern, U. Franke,U.-P. Kaeppler e P. Levi: Detecting Reflection Posts - Lane Recognition on Country Roads *IEEE Intelligent Vehicles Symposium*, 14-17 giu. 2004.
- [7] R. Cucchiara, M. Piccardi e P. Mello: Image Analysis and Rule-Based Reasoning for a Traffic Monitoring System. *IEEE Transaction on Intelligent Transportation System*, vol. 1(2):pp. 119-130, giu. 2000.

-
- [8] R. DeFauw, S. Lakshmanan e K.V. Prasad: A System for small Target Detection, Tracking, and Classification. *IEEE/IEEJ/JSAI International Conference on Intelligent Transportation Systems Proceedings*, 5-8 ott. 1999
- [9] C. E. Smith, C. A. Richards, S. A. Brandt; N. P. Papanikolopoulos: Visual Tracking for Intelligent Vehicle-HighWay System. *IEEE Transaction on Vehicular Techonology*, vol. 45(4):pp. 744-759, nov. 1996.

Ringraziamneti

Innanzitutto desidero ringraziare il Prof. Alberto Broggi e il Dott. Nereo Pallaro per avermi dato l'opportunità di partecipare a questa esperienza.

La collaborazione con la Divisione Tecnologie Ottica e Fotonica del Centro Ricerche Fiat mi ha dato l'occasione di confrontare le metodologie studiate nel corso degli anni con una vera applicazione industriale e commerciale. Credo di aver dato tutto quello che era nelle mie capacità con puntuale precisione e credo anche che questo sia stato apprezzato. Debbo inoltre ringraziare tutti i membri del Laboratorio di Visione Artificiale che hanno, in modi diversi, contribuito alla realizzazione di questa Tesi, con una particolare menzione al correlatore, l'Ing Pietro Cerri, che ha pazientemente letto, e corretto, il mio lavoro. Come non ricordare però, l'Ing. Paolo Grisleri, l'Ing. Paolo Medici, l'Ing. Stefano Ghidoni e i futuri Ing. L. Mussi, M. Maccari, F. Pellegrini e N. Bernini. Lavorare con loro è stata una esperienza certamente importante ed istruttiva, e che mi sarà di esempio, nel bene e nel male.

Un ringraziamento molto particolare va ai miei genitori, per avermi permesso di raggiungere questo obiettivo nel migliore dei modi e per il costante supporto economico ed emotivo datomi. Un bacione grosso ad Annalisa, per la pazienza, la comprensione e il grande affetto dimostratomi in questi ultimi anni di Università. Queste sono cose che non si dimenticano e restano lì, in fondo al cuore, per sempre.

Un cinque ed un grazie a Marco, Mario e Rita, Beckam, Alex, Guti, Gauss, Mauri, Carlo e Gabri e Mostrino, Vecchio e Lorena, Androlone, Jhonny, Rosanna e Beppe, per le belle serate, le grandi partite a pallone, i viaggi fatti insieme che mi hanno opportunamente tenuto lontano dai libri, e che hanno salvaguardato la mia salute mentale. Infine un grazie a tutti quelli che ho dimenticato. A tutti quelli di cui non ricordo più il

nome, o di cui non l'ho mai saputo, ma che hanno in qualche modo incrociato la mia vita di studente durante tutti questi anni.

Ragazzi, ho finito, quando andiamo da Sincero? Ho fame!