



**UNIVERSITÀ DEGLI STUDI DI PARMA**  
DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

*Dottorato di Ricerca in Tecnologie dell'Informazione*  
*XX Ciclo*

Claudio Caraffi

**STUDIO E REALIZZAZIONE  
DI SISTEMI DI PERCEZIONE FRONTALE  
PER AUTOVEICOLI  
MEDIANTE VISIONE ARTIFICIALE**

DISSERTAZIONE PRESENTATA PER IL CONSEGUIMENTO  
DEL TITOLO DI DOTTORE DI RICERCA

GENNAIO 2008



*Alla Fortezza Bastiani*





<b>3</b>	<b>Rilevazione delle corsie stradali</b>	<b>27</b>
3.1	Immagine del giallo . . . . .	27
3.2	Elaborazione di basso livello . . . . .	29
3.2.1	Calcolo dell'IPM e estrazione dei pattern DLD . . . . .	29
3.2.2	Mascheratura degli ostacoli . . . . .	32
3.2.3	Etichettatura (Labeling) . . . . .	33
3.2.4	Rimozione dal b/n dei label del giallo . . . . .	34
3.2.5	Conversione a coordinate mondo . . . . .	34
3.3	Creazione del modello stradale . . . . .	37
3.3.1	Individuazione della linea madre e creazione delle linee spezzate . . . . .	37
3.3.2	Segmentazione . . . . .	41
3.3.3	Curve . . . . .	41
3.3.4	Linee tratteggiate . . . . .	44
3.3.5	Tipi di linea . . . . .	46
3.3.6	Guardrail . . . . .	46
3.4	Tracking . . . . .	50
3.5	Conclusioni e sviluppi futuri . . . . .	52
<b>4</b>	<b>Determinazione del profilo del terreno</b>	<b>55</b>
4.1	Estrazione del profilo del terreno . . . . .	56
4.2	Stirare la superficie stradale . . . . .	61
4.3	Risultati e conclusioni . . . . .	63
<b>II</b>	<b>Percezione di segnali stradali</b>	<b>65</b>
	<b>Presentazione del problema</b>	<b>67</b>
<b>5</b>	<b>Rilevazione di cartelli di fine prescrizione</b>	<b>69</b>
5.1	Individuazione di bande nere verticali inclinate . . . . .	70
5.1.1	Ricerca di campioni High-Low-High . . . . .	71

---

5.1.2	Etichettatura . . . . .	76
5.1.3	Validazione e unione di regioni etichettate . . . . .	77
5.2	Individuazione di cerchi . . . . .	79
5.3	Tracking . . . . .	87
5.4	Risultati . . . . .	93
5.5	Considerazioni . . . . .	95
5.6	Conclusioni . . . . .	96
<b>6</b>	<b>Riconoscimento di pannelli integrativi</b>	<b>99</b>
6.1	Caratteristiche dei pannelli integrativi e scelta delle regioni di ricerca	100
6.2	Utilizzo della visione stereo . . . . .	101
6.3	Elaborazione delle funzioni relative ai bordi . . . . .	104
6.3.1	Elaborazione dei bordi verticali . . . . .	104
6.3.2	Elaborazione dei bordi orizzontali . . . . .	111
6.3.3	Unione degli istogrammi e validazione . . . . .	114
6.4	Confronto delle intensità luminose . . . . .	115
6.5	Tempi computazionali . . . . .	118
6.6	Risultati e conclusioni . . . . .	118
	<b>Conclusioni</b>	<b>121</b>
	Epilogo . . . . .	122
	<b>Appendici</b>	<b>123</b>
<b>A</b>	<b>V-Disparity Image Analysis</b>	<b>125</b>
A.1	Computation . . . . .	126
A.2	Quality and road flatness measurement . . . . .	129
A.3	Discussion . . . . .	131
<b>B</b>	<b>Obstacle Detection</b>	<b>133</b>
B.1	Algorithm overview . . . . .	133

---

B.2	DSI computation . . . . .	134
B.3	Obstacles search in the DSI . . . . .	135
B.4	Real world coordinates mapping . . . . .	140
B.5	Obstacle detection results . . . . .	142
<b>C</b>	<b>Disegno di linee su immagini</b>	<b>143</b>
	<b>Bibliografia</b>	<b>145</b>





# Introduzione

Nell'anno 2005, sulle strade dei 25 stati membri dell'Unione Europea all'epoca<sup>1</sup>, sono stati registrati oltre un milione di incidenti con feriti<sup>2</sup> che hanno causato oltre 40 mila vittime [4]. Dai dati riportati in tabella 1, è possibile osservare come sia il numero di incidenti sia soprattutto il numero di vittime della strada siano in diminuzione.

La sicurezza stradale ha fatto significativi progressi, soprattutto negli anni più recenti. Questo è il risultato del migliorato design dei veicoli, di infrastrutture rese più sicure e soprattutto di un maggiore rispetto del codice della strada da parte dei guidatori. L'applicazione più rigida di misure che erano già prescritte da molto tempo dal codice della strada, per esempio riguardo a limiti di velocità, abuso di alcol e uso delle cinture di sicurezza, spiega in buona parte i miglioramenti della sicurezza stradale. Allo stesso modo, è plausibile che il progresso sia anche dovuto ad un comportamento più consapevole di molti automobilisti. Le politiche repressive e di educazione hanno innegabilmente dato frutto, e devono perciò essere conservate e sostenute. Ciò nonostante, è probabile che la maggior parte del progresso che era possibile ottenere tramite questi strumenti sia già stato conseguito [6], quando ancora il prezzo in vite umane e in feriti pagato è troppo alto.

Solo una piccola percentuale degli incidenti è ormai inputabile a guasti meccanici

---

<sup>1</sup>Con l'adesione di Romania e Bulgaria, dal 1 gennaio 2007 gli stati membri dell'Unione Europea sono diventati 27.

<sup>2</sup>Nel resto della trattazione nelle statistiche con il termine "incidente" si intenderà un evento che abbia causato feriti.

Anno	EU-15		EU-25		Italia	
	Incidenti	Decessi	Incidenti	Decessi	Incidenti	Decessi
1970	1 388 500	77 831	n/d	n/d	173 100	11 004
1980	1 379 500	64 237	n/d	n/d	163 800	9 220
1990	1 293 500	56 413	1 422 100	n/d	161 800	7 137
2000	1 286 200	40 761	1 417 600	52 241	256 546	7 061
2001	1 281 400	39 849	1 412 600	50 384	263 100	7 096
2002	1 240 000	38 604	1 304 400	49 719	265 402	6 980
2003	1 208 500	37 253	1 344 400	46 756	252 271	6 563
2004	1 167 500	32 520	1 304 100	43 355	243 490	6 122
2005	1 132 900	30 959	1 255 500	41 274	240 011	5 818
2006	n/d	n/d	n/d	n/d	238 124	5 669

Tabella 1: Statistiche sugli incidenti stradali.

EU-15 indica l'insieme dei 15 stati membri dell'Unione Europea prima dell'allargamento a 25 stati (EU-25) del 1 maggio 2004. Tutti i dati sono tratti dal rapporto ERF del 2007 [4], ad eccezione delle statistiche per l'Italia dal 2000 al 2006, la cui fonte è il rapporto ISTAT sugli incidenti stradali 2006 [5].

(in Italia, lo 0,2%). Tra gli altri, airbag, dispositivi di frenata ABS e i più recenti sistemi di stabilizzazione ESP hanno permesso di ridurre il numero e gli effetti degli incidenti. La tecnologia applicata ai veicoli può però ancora fare molto, e cresce a livello mondiale l'interesse per i sistemi di sicurezza che prevedano il crearsi di situazioni di pericolo causate da imprudenza, stanchezza o distrazione del guidatore, con la possibilità quindi di richiamare l'attenzione o di intervenire direttamente in modo automatico sulla guida. Fondamentale per la realizzazione di questo scopo è la percezione, senza la quale il sistema non può avere le informazioni per prendere le decisioni di intervento. E' in questo contesto che si colloca questa tesi, con la presentazione nella prima parte del sistema di percezione frontale realizzato per i veicoli automatici TerraMax e TerraMax2, e con la descrizione nella seconda parte di due algoritmi per la rilevazione di segnaletica verticale.

La prima osservazione che è necessario fare è che un sistema di percezione per un sistema di guida automatica o assistenza alla guida deve avere un tasso di errore (mancata percezione o falsa percezione) molto basso, se non nullo: chi affiderebbe la propria vita a un congegno di guida automatica che *una volta su un milione* sbaglia a rilevare la corsia e porta il veicolo fuori strada? O quanto impiegherebbe l'autista a smettere di fidarsi in un sistema di rilevazione di cartelli stradali, qualora le informazioni ricevute fossero sbagliate anche solo una volta su mille?

Un altro punto importante è che una fetta significativa degli studi di visione artificiale sono interessati allo studio di algoritmi dai risultati sempre più precisi, ma senza particolare attenzione ai tempi di elaborazione. Questo può trovare applicazione in molti campi (analisi di immagini mediche diagnostiche per esempio), ma non di certo per il caso in esame.

I sistemi di percezione che saranno presentati sono stati pertanto realizzati con un occhio di riguardo ai tempi di computazione; saranno inoltre tratti i bilanci dei risultati conseguiti per i singoli sistemi, con un'esame dei limiti ancora presenti, e di quali problemi ancora è necessario trovare soluzioni per poter avere finalmente un'applicazione che possa, infine, salvare delle vite.

## Sommario

Nella prima parte della tesi sarà presentato il sistema di percezione frontale a tre telecamere implementato per i veicoli autonomi TerraMax e TerraMax2 (fig. 1) che hanno partecipato alle edizioni della DARPA<sup>3</sup> Grand Challenge 2004 e 2005, e Urban Challenge 2007.



Figura 1: I veicoli TerraMax e TerraMax2

La DARPA Grand Challenge è una gara a cronometro per veicoli autonomi su un percorso di oltre 200 km nel deserto con tratti di collina, su strade per oltre l'80% non asfaltate. Dopo il sostanziale fallimento del 2004 (il miglior veicolo si fermò dopo 12 km, il TerraMax dopo 2), l'edizione dell'8 ottobre 2005 ha visto giungere al traguardo ben 5 veicoli, tra cui il TerraMax, unico a basarsi anche su visione artificiale e non solo su laserscanner per la rilevazione di ostacoli.

In seguito a questi successi, con la Urban Challenge la sfida è stata spostata in città: i veicoli partecipanti hanno dovuto cimentarsi seguendo il codice della strada e, soprattutto, interagire con altri veicoli (autonomi o guidati da uomini dell'organizzazione). Dopo una buona qualificazione, in gara il TerraMax2 è stato fermato dai giudici di gara mentre si muoveva verso un edificio in linea retta senza controllo alla velocità di 1 mph, a causa di un errore del sistema di pianificazione del moto che aveva messo in crisi il sistema di attuazione.

<sup>3</sup>Defense Advanced Research Project Agency

---

Il capitolo 1 presenta il sistema di acquisizione i principi alla base del suo design e il problema della calibrazione, con un'introduzione riguardo alla V-Disparity image, alla base di alcuni degli algoritmi presentati. Nel capitolo 2 sarà descritto un metodo per la stima del punto di fuga al fine di misurare un angolo di calibrazione cruciale per il funzionamento del sistema: lo yaw. Il capitolo 3 presenterà il sistema di percezione delle corsie implementato per il veicolo TerraMax2, mentre con il capitolo 4 è mostrato un metodo per l'estrazione del profilo del terreno e la rimozione dell'effetto della pendenza stradale sulle immagini. Inerentemente a questi argomenti, l'appendice A riporta (in lingua inglese) uno studio della V-Disparity image per stabilizzazione di immagini in ambiente on-road e off-road, stima della planarità del terreno e valutazione della qualità delle immagini acquisite. Infine in appendice B (in lingua inglese) è descritto l'algoritmo di percezione ostacoli basato su visione stereo utilizzato sul veicolo TerraMax<sup>4</sup> durante l'edizione 2005 della DARPA Grand Challenge.

Nella seconda parte della tesi sono presentati due algoritmi, inerenti alla percezione dei cartelli stradali, frutto della proficua e stimolante collaborazione con l'(allora non) Ing. Elena Cardarelli, che qui ringrazio. Nel capitolo 5 è trattata la rilevazione dei segnali di fine prescrizione, caso particolare nel panorama dei cartelli stradali perché privo di colore. Uno studio per la rilevazione dei pannelli integrativi associati ai segnali stradali è infine esposto nel capitolo 6, al termine del quale si potranno trovare alcune brevi conclusioni generali, derivate dalla mia esperienza nel campo della visione artificiale in questi ultimi 4 anni.

Gli algoritmi presentati sono stati sviluppati in C++ all'interno del framework GOLD (Generic Obstacle and Lane Detection) del VisLab, che ha messo a disposizione funzioni di libreria per l'acquisizione e visualizzazione di immagini, per il tuning online delle soglie tramite pannello e per le trasformazioni geometriche (triangolazione e inverse perspective mapping).

Prima di procedere, vorrei portare i miei doverosi ringraziamenti al mio tutor, il Prof. Alberto Broggi. Un grazie sincero va a tutti i ragazzi (compreso Massimo :) ) del VisLab e della palazzina. Con voi questi anni non sono stati solo di ricerca e lavoro,

---

<sup>4</sup>in versione rivista e aggiornata è stato impiegato anche sul veicolo TerraMax2

ma un'avventura vissuta ai confini delle zone inesplorate della Terra: chi aveva mai sentito parlare di Cedar Rapids, Oshkosh, Marquette, Appleton, Carlson City e via dicendo... Per non menzionare Scicli (RG)! Spero conserverete un buon ricordo di me come io di voi!

Un grazie va a tutti i miei amici... Stavolta non mi dilungherò come in passato [7], ma passerò ai fatti: *“Allora, quand'è che usciamo a festeggiare?!?”* :) Grazie alla stupendamente raffazzonata compagnia GSM di Toledo e al gruppo del teatro con i quali ho vissuto esperienze straordinarie (ho anche imparato a fare passing e sputare fuoco, ma questa è un'altra storia...).

Mi rimangono due grazie speciali da recapitare.

Il primo è per la mia famiglia, per il sostegno, per il confronto onesto e per la meraviglia di essere zio!

Il secondo è per l'unica Radha italiana al 100%, che è solo mia! Un grazie infinito per questo tempo di gioia e serenità mai provate!

## **Parte I**

# **Sistema di percezione frontale per un autoveicolo autonomo**



## Capitolo 1

# Il sistema di percezione frontale dei veicoli TerraMax

Un veicolo automatico deve essere dotato di 3 capacità fondamentali:

- **Sensoriale:** attraverso strumenti come GPS, laserscanner, radar, visione il sistema deve essere in grado di percepire l'ambiente circostante e ricostruirlo.
- **Pianificazione:** raccolti i dati sensoriali e valutati gli obiettivi, il sistema deve decidere le azioni da compiere, il percorso da seguire per dirigersi verso la destinazione e le reazioni da apportare per reagire ad imprevisti.
- **Attuazione:** le istruzioni del livello di pianificazione devono essere attuate su sterzo, acceleratore, cambio e freno.

Anche se un livello non può prescindere da un altro (è infatti assolutamente indispensabile un dialogo tra chi sviluppa la sensorialità e chi la pianificazione per conoscere quali sono i dati attesi e le informazioni sulle quali è bene concentrare l'attenzione), in questa tesi sarà trattato il solo problema della percezione, e in particolare della percezione frontale, ovvero nella direzione del moto del veicolo.

## 1.1 Visione Stereo e sistema trinoculare

Quando si trovano alla guida, gli esseri umani fanno scarso affidamento sulla visione stereo<sup>1</sup>: utilizzando l'analisi del moto e l'esperienza accumulata, siamo infatti in grado di guidare tranquillamente con un occhio chiuso. Questa capacità è però difficilmente riproducibile, soprattutto in ambiente off-road. Le capacità di ricostruzione stereo possono però essere potenziate utilizzando occhi, e in questo caso telecamere, posti ad una distanza tra loro maggiore: questa misura è detta *baseline*, e la scelta in fase di progetto del sistema di percezione mediante visione artificiale dei veicoli TerraMax e TerraMax2 è stato di utilizzare appunto baseline ampie, fino cioè a 1,5 m.

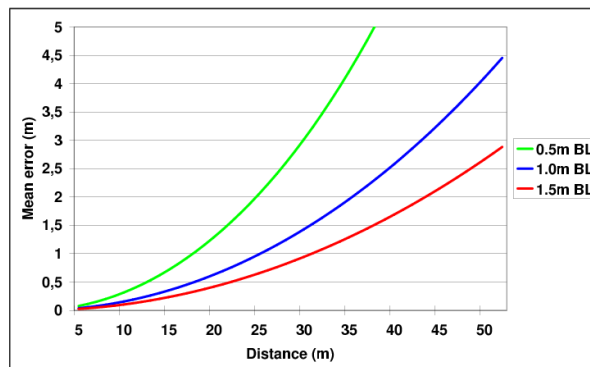


Figura 1.1: Stima dell'errore medio nella misura della distanza in funzione della distanza e della baseline (BL) utilizzata, considerando il sistema in uso sul TerraMax.

In assenza di conoscenze a priori, baseline ampie permettono un'accurata stima della distanza anche per oggetti particolarmente distanti. Nel grafico di figura 1.1 è possibile osservare come l'errore di stima della distanza sia in dipendenza appunto

<sup>1</sup>Per una trattazione della visione stereo può essere fatto riferimento al primo capitolo della mia tesi di laurea [7].

della distanza e in proporzione inversa rispetto alla misura della baseline utilizzata. Tuttavia, per oggetti vicini le due telecamere forniscono un angolo di vista molto diverso rendendo l'individuazione di punti omologhi difficile. Per questa ragione è stato implementato un sistema a 3 telecamere in posizione asimmetrica fissate alla stessa barra che permette di selezionare due telecamere ottenendo una baseline corta (0,5 m), media (1,0 m) o lunga (1,5 m) a seconda delle necessità (fig. 1.2).



Figura 1.2: Una foto ravvicinata del sistema a 3 telecamere del TerraMax. Il sistema del TerraMax2 è simile, con le telecamere montate in corrispondenza del punto più alto del parabrezza all'interno dell'abitacolo.

In particolare la selezione è stata implementata in funzione della velocità: per alte velocità il range visivo è spostato il più in là possibile utilizzando la baseline più larga, mentre la baseline corta è utilizzata quando il veicolo si muove lentamente.

Un sistema di questo tipo risente in modo significativo del problema della calibrazione. Per calibrazione di un sistema di visione possono essere intesi due significati:

- L'insieme dei dati misurati per i parametri intrinseci (apertura orizzontale e frontale, posizione del centro dell'immagine) ed estrinseci (posizione e orientazione, si veda la figura 1.3) delle telecamere.
- Le operazioni di installazione delle telecamere, con verifica del corretto allineamento per un sistema stereo e accorgimenti per evitare di avere immagini in qualche modo ruotate.

Entrambi gli aspetti sono molto importanti: occorre prestare la massima attenzione durante la fase di fissaggio delle telecamere, per fare in modo che le telecamere siano il più possibile vicini ad una forma standard (perfetto allineamento per tutti gli angoli del piano delle immagini, relativamente alla direzione della congiungente le

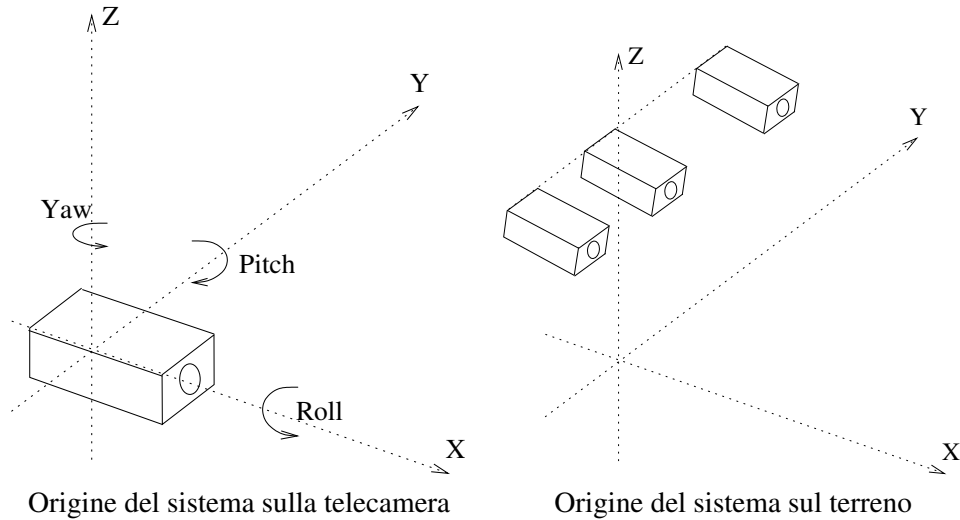


Figura 1.3: Disposizione degli assi e angoli di rotazione.

due telecamere), in modo che il calcolo stereo parta da una situazione di vantaggio. Il fissaggio delle telecamere deve essere molto stabile, in quanto un movimento delle telecamere dovuto alle sollecitazioni (consistenti in ambiente off-road) può vanificare tutti gli sforzi di scrittura di un buon algoritmo di ricostruzione stereo. Inoltre, la raccolta di precise misure di calibrazione (nel primo significato del termine) consente di calcolare un insieme di vincoli sulla corrispondenza tra punti dell'immagine e punti del mondo reale.

Nella trattazione della V-Disparity Image e nel capitolo 2 saranno esaminati alcuni metodi attraverso i quali è possibile conoscere se la calibrazione è stata effettuata in modo corretto, con un'indicazione sulle correzioni da mettere in atto. Quello che è inoltre possibile affermare in generale è che la presenza di altri sensori sul veicolo costituisce un elemento di cui tener conto in fase di calibrazione: laserscanner e visione devono avere valori di calibrazione che permettano una corrispondenza perfetta tra gli elementi trovati. Inoltre, attraverso test sperimentali sul comportamento autonomo del veicolo ad esempio associato all'output dell'algoritmo di rilevazione delle

corsie, è possibile constatare un'eventuale errore di calibrazione qualora sia mandato in uscita una corsia che spinga il camion a cercare continuamente di uscire da essa.

Il sistema trinoculare è stato utilizzato nel veicolo TerraMax per la rilevazione di ostacoli e di un percorso sicuro, mentre nel veicolo TerraMax sempre per la rilevazione di ostacoli e per l'individuazione di corsie stradali.

In appendice B è descritto l'algoritmo di percezione ostacoli utilizzato dal TerraMax durante la DARPA Grand Challenge 2005<sup>2</sup> Il capitolo 3 descrive invece l'algoritmo di percezione delle corsie utilizzato sul veicolo TerraMax2 durante le fasi delle DARPA Urban Challenge 2007.

Chiudo il capitolo con una digressione sulla V-Disparity Image, strumento che sarà di base per l'estrazione di informazioni utili a tutti gli algoritmi presentati in questa prima parte della tesi.

## 1.2 V-Disparity Image

Avendo a disposizione una coppia stereo di immagini è possibile calcolare la corrispondente V-Disparity Image. Il procedimento consiste nel dividere la coppia stereo in righe orizzontali di pixel, e di misurare tra righe dell'immagine di sinistra e righe dell'immagine di destra prese alla stessa altezza quali siano i "valori di somiglianza" (correlazione) per le due righe in funzione della disparità di confronto. E' possibile riportare in un grafico tridimensionale i valori di somiglianza ottenuti in funzione dell'altezza nell'immagine (coordinata  $v$ ) e della disparità (coordinata  $d$ ) per la quale viene eseguita la misura; come terza dimensione si utilizza l'intensità luminosa, che quindi sarà proporzionale al valore di somiglianza misurato. Questa rappresentazione costituisce appunto la V-Disparity image 1.4.

### 1.2.1 Proprietà

Nella V-Disparity image di esempio sono visibili due segmenti verticali:

---

<sup>2</sup>Per la Urban Challenge 2007 è stata utilizzata una versione dell'algoritmo simile, adattata per l'ambiente cittadino e migliorata in alcuni aspetti.

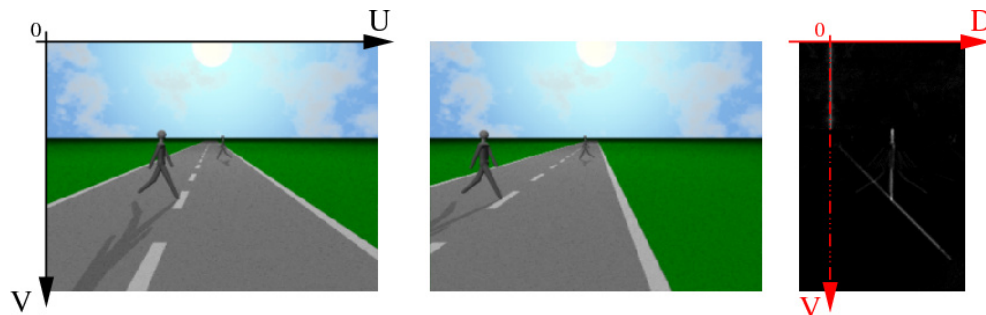


Figura 1.4: V-Disparity Image. La coordinata  $v$  dell'immagine è spostata a destra di un valore costante per permettere di rappresentare anche disparità negative, l'origine dell'immagine risulta quindi anch'essa spostata a destra.

- il primo, parzialmente nascosto dall'asse  $v$ , rispecchia un'area dell'immagine per cui il massimo valore di somiglianza è ottenuto a disparità nulla. La regione corrisponde quindi a punti posti a distanza infinita, ovvero il cielo, delimitato dall'orizzonte.
- il secondo, posto a mezza altezza, è originato dalla presenza del pedone, che causa un buon abbinamento delle linee dell'immagine per il valore di disparità che lo contraddistingue. Infatti, l'ostacolo individua con buona approssimazione una superficie parallela al piano delle telecamere, contraddistinta quindi da una disparità costante.

Quello che è possibile ricavare [8] è che ogni superficie reale approssimativamente piana e parallela all'asse congiungente le 2 telecamere costituenti il sistema stereo di acquisizione si manifesta nella V-Disparity image con un segmento. Per quanto riguarda superfici perpendicolari al terreno (quindi gli ostacoli), tale segmento è perfettamente verticale se le telecamere sono perfettamente orizzontali<sup>3</sup>, in quanto un unico piano perpendicolare agli assi ottici taglia la superficie visibile dell'oggetto che ha quindi disparità costante nelle immagini.

<sup>3</sup>I loro assi ottici sono cioè paralleli al terreno

Oltre al pedone, una seconda superficie, costituita dal terreno, è presente nell'immagine. Anch'essa in questo caso è piana e parallela all'asse congiungente le telecamere, perciò genera nella V-Disparity image un segmento. Tale segmento non è però verticale, ma inclinato, in quanto la disparità del terreno diminuisce al crescere della distanza. La linea inclinata (e in questo caso retta) visibile nella V-Disparity Image rappresenta la **linea di correlazione del piano stradale**, o più brevemente linea di correlazione.

Se il terreno ripreso nelle immagini presenta una pendenza, ma è comunque approssimabile con una serie di superfici piane e parallele all'asse congiungente le telecamere, la linea di correlazione del piano stradale risulta essere non più retta ma piegata, come è possibile vedere nell'esempio di figura 1.5.

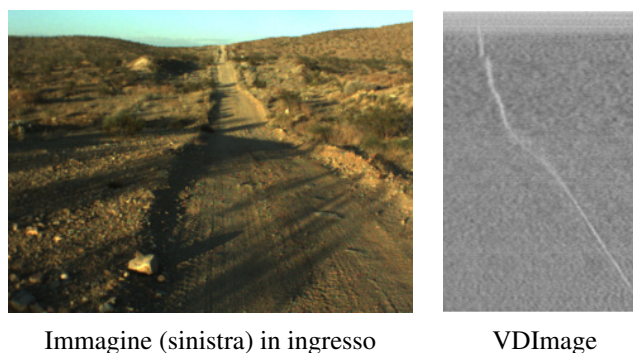


Figura 1.5: Effetto di una pendenza sulla V-Disparity image.

### 1.2.2 Impieghi

Come già anticipato, il contenuto informativo della V-Disparity image è di fondamentale importanza per il funzionamento di tutti i sistemi di percezione presentati in questa prima parte della tesi e nelle appendici.

### 1.2.2.1 Stima della posizione del terreno

Un sistema di acquisizione montato su un veicolo subisce inevitabilmente oscillazioni dovute ad accelerazioni, frenate e alle irregolarità del terreno; come è facile pensare, in ambiente fuori strada l'effetto è accentuato. L'individuazione della linea di correlazione sulla V-Disparity image, in prima approssimazione modellizzata con una retta, consente di conoscere la posizione del piano stradale rispetto alle telecamere all'istante di acquisizione, e quindi di ricostruire in tempo reale l'entità dell'oscillazione del sistema. Questa informazione è di fondamentale importanza per effettuare correttamente le trasformazioni geometriche necessarie agli altri algoritmi, per esempio:

- **triangolazione** per la ricostruzione della posizione nel mondo degli oggetti nel caso della rivelazione di ostacoli;
- **rimozione della prospettiva** per il sistema di riconoscimento delle corsie stradali.

La posizione del terreno è inoltre un vincolo che permette di ridurre i gradi di libertà della grandezza in osservazione.

- nel calcolo della DSI (Disparity Space Image), necessaria per l'individuazione di ostacoli (appendice B), per ogni riga dell'immagine è possibile ridurre il numero di disparità esaminate spostando il range di ricerca in base alla disparità del terreno attesa per la riga stessa;
- nel caso del calcolo di un percorso fuori strada sicuro [9], la retta di correlazione contiene l'informazione sulla posizione dell'orizzonte e perciò consente di individuare una regione di interesse dell'immagine; inoltre la conoscenza della posizione dell'orizzonte consente di stabilire su quale riga dell'immagine ricercare il punto di fuga (capitolo 2).

Un minor numero di gradi di libertà per la grandezza osservata comporta anche un minor numero di casi da esplorare, con una riduzione dei possibili errori e, fatto da non trascurare, dei tempi computazionali.

### 1.2.2.2 Informazioni sulla pendenza del terreno

Come presentato in figura 1.5, la presenza di una pendenza di fronte al sistema di acquisizione si manifesta nella V-Disparity image calcolata dalle immagini in ingresso con la comparsa di una linea di correlazione non retta. L'analisi in questo caso permette importanti impieghi:

- La semplice constatazione della presenza di una pendenza, può indurre gli altri algoritmi che devono utilizzare le immagini a fare considerazioni più prudenti, diminuendo l'affidabilità data al proprio risultato o utilizzando un set di parametri (soglie) più restrittivi.
- La raccolta di semplici statistiche permette inoltre di fare fronte in modo più mirato ai problemi causati, ad esempio, nel caso della rilevazione di ostacoli, contesto nel quale un cambio di pendenza può causare falsi positivi.
- Infine, uno studio più approfondito può permettere di ricavare in modo completo la linea di correlazione del piano stradale, ottenendo quindi una stima del profilo del terreno.

### 1.2.2.3 Controllo della qualità delle immagini e dell'allineamento delle telecamere

La V-Disparity image contiene infine preziose informazioni riguardo alla qualità delle immagini acquisite: valutando infatti la nitidezza della linea di correlazione del terreno è infatti possibile capire se le immagini in ingresso sono in una certa misura deteriorate, prendendo eventualmente provvedimenti per esempio nella scelta dei parametri da parte degli algoritmi che le devono utilizzare. Infine, sfruttando lo stesso principio, è possibile avere informazioni riguardo allo stato della calibrazione delle telecamere, e in particolare riguardo alla precisione del loro allineamento, caratteristica indispensabile per un corretto funzionamento del sistema.

### 1.2.3 Trattazione

L'appendice A (in lingua inglese) ha inizio con la presentazione del metodo implementato per il calcolo della V-Disparity image, che si è dimostrato essere efficace anche in ambienti difficili come quelli fuori strada e dotato di tempi computazionali ridotti. La trattazione prosegue con il metodo di calcolo della retta di correlazione e dell'oscillazione istantanea del sistema. E' inoltre spiegato il metodo di verifica di presenza di terreno piatto e il sistema di controllo della qualità dell'immagine e dello stato dell'allineamento delle telecamere.

L'appendice B (in lingua inglese) tratta il sistema di ricerca degli ostacoli, e al suo interno spiega l'utilizzo delle informazioni della V-Disparity image per la riduzione del range di ricerca di disparità e per la rimozione di falsi positivi causati da cambi di pendenza.

Il capitolo 4 presenta infine uno studio sulla V-Disparity image per l'estrazione di un profilo completo di pendenze del terreno e per una curiosa operazione che questa informazione rende possibile: la stiratura delle pendenze nelle immagini.

## Capitolo 2

# Punto di fuga e angolo di yaw

In prospettiva, il punto di fuga (in inglese *vanishing point*) è il punto verso il quale sembra convergere una serie di linee che sono in realtà parallele fra loro. Nell'immagine in figura 2.1, il punto di fuga descritto della strada è individuabile ricavando il punto di intersezione delle linee in essa contenute (segnaletica orizzontale, marciapiedi, tracce di altri veicoli...). Lo scopo di questa analisi è in primo luogo l'individuazione del punto di fuga della strada nell'immagine.



Figura 2.1: Esempio di punto di fuga.

Per il processo di calibrazione, ponendo il veicolo su una strada perfettamente piana e rettilinea in posizione perfettamente parallela alla strada, conoscendo i parametri

intrinseci della telecamera e ricavando il punto di fuga della strada nell'immagine è possibile risalire agli angoli di pitch e yaw della calibrazione [10]. Come è ovvio pensare, la situazione ipotizzata non è riproducibile nella realtà, soprattutto per quanto riguarda il posizionamento corretto del veicolo. Quello che è proposto in questo studio è una stima della posizione del punto di fuga nelle condizioni citate mediante l'analisi di una sequenza di immagini, partendo dal presupposto che *in media* il veicolo segue una direzione parallela alla strada, che *in media* è piana e rettilinea.

## 2.1 Stima del punto di fuga

Una prima premessa che è possibile fare è che, essendo a conoscenza di una prima stima degli angoli di yaw, nel caso di immagini stereo tramite l'analisi della V-Disparity image è possibile fin da subito conoscere una stima del pitch e conseguentemente della posizione dell'orizzonte dell'immagine (par. 1.2) sapendo quindi su quale linea dell'immagine il punto di fuga si troverà. Un punto di fuga candidato sarà quindi definito dalla sola coordinata  $u$  nell'immagine, in quanto la coordinata  $v$  sarà data dalla stima della posizione dell'orizzonte. In base alla precisione con la quale si vuole stimare la posizione del punto di fuga, è quindi stabilito un set di punti di fuga candidati (punti gialli dell'immagine 2.1).

Prendendo ora in considerazione l'immagine ternarizzata 2.2 è possibile osservare come questa evidenzi delle regioni di colore omogeneo convergenti verso il punto di fuga. Scegliendo arbitrariamente un set di punti alla base dell'immagine, è ragionevole ipotizzare che lungo le linee congiungenti i punti appartenenti a questo set e i possibili punti di fuga, il numero di cambi di colore sia tanto basso quanto più il punto di fuga candidato è vicino al punto di fuga reale. L'algoritmo realizzato conteggia quindi per ogni punto di fuga candidato i cambi di segno lungo tutte le congiungenti con i punti scelti alla base dell'immagine.

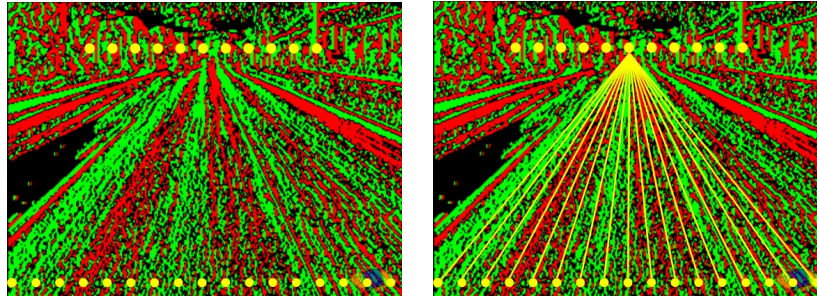


Figura 2.2: Immagine ternarizzata (ottenuta dall'immagine di fig. 2.1), punti candidati ad essere punto di fuga e punti utilizzati per tracciare le linee necessari per eseguire il conteggio delle variazioni di segno. L'immagine ternarizzata, introdotta per estrarre informazioni dal piano stradale anche in ambienti poco strutturati (si veda l'app. A), rappresenta il segno dell'immagine filtrata secondo un'operatore di Sobel verticale: scorrendo le righe dell'immagine originale verso destra e valutando la variazioni locali di tono di grigio, nell'immagine ternarizzata il rosso indica una variazione negativa, mentre il verde indica una variazione positiva.

## 2.2 Realizzazione

L'algoritmo, concettualmente molto semplice, ha richiesto una piccola modifica che si è dimostrata necessaria dopo i primi test sperimentali: la regione alla base dell'immagine e la regione vicino all'orizzonte (quest'ultima a meno di strade perfettamente rettilinee e sgombre), presentano di norma un contributo di rumore eccessivo che si è deciso di scartare concentrando il conteggio dei cambi di colore in una fascia orizzontale dell'immagine che escludesse appunto queste zone di rumore. E' stata inoltre posta una soglia sul numero di cambi di colore massimi lungo una linea, oltre la quale il contributo della linea è considerato saturato: in questo modo si è cercato di privilegiare l'apporto di linee contenenti informazione e non solo rumore.

Il punto di fuga per un'immagine è quindi calcolato come quello tra i candidati che ha conteggiato il minor numero di cambi di colore (nelle figure punto nero sulla linea dell'orizzonte). In una sequenza di immagini, il numero di cambi di colore per ogni

frame sono inoltre accumulate dai punti di fuga candidati, permettendo quindi di ottenere il punto di fuga mediato su una sequenza (*punto di fuga storico*, nelle figure punto bianco sulla linea dell'orizzonte), obiettivo di questo studio.

### 2.3 Risultati e impieghi

L'algoritmo realizzato converge rapidamente e stabilmente al punto di fuga corretto della sequenza di immagini, anche in caso di oscillazioni del veicolo. Esaminando l'esempio di figura 2.3, è possibile osservare come l'algoritmo sia in grado di comportarsi correttamente anche quando la sequenza comprende tratti di percorrenza su strada non piana (fig. 2.3a), curva (fig. 2.3b, 2.3c) o avente tessitura molto casuale (fig. 2.3d): non appena il veicolo affronta una strada rettilinea (fig. 2.3e) il contributo di informazione accumulato prevale rapidamente sul rumore introdotto nei frame precedenti (fig. 2.3f). Il metodo presentato può essere quindi utilizzato per misurare l'angolo di calibrazione yaw delle telecamere utilizzate per l'acquisizione.

Gli angoli stimati in questo modo possono quindi essere utilizzati per ripetere l'operazione consentendo una stima più precisa per ogni frame del pitch<sup>1</sup> e quindi della posizione dell'orizzonte nell'immagine, arrivando per iterazione ad una stima più precisa della calibrazione. Tuttavia questo metodo richiede una verifica sperimentale, attualmente non ancora attuata.

Essendo pensato per una calibrazione a freddo del sistema, cioè su sequenza di immagini o comunque senza altre elaborazioni in corso sull'elaboratore, la velocità di elaborazione dell'algoritmo non è considerata di primaria importanza. Comunque, i tempi ottenuti per l'elaborazione del punto di fuga su un sistema a processore P4@2.8GHz su immagini  $320 \times 240$  sono stati di 75ms per immagine considerando 160 punti di fuga, e di 150ms considerando tutti i 320 possibili punti di fuga.

---

<sup>1</sup>I calcoli al primo passo si avvalgono di yaw stimati, e possono perciò essere resi più precisi introducendo i nuovi valori di yaw ottenuti.



Figura 2.3: Individuazione del punto di fuga in una sequenza. Il cerchio bianco indica il punto di fuga storico, il cerchio nero più piccolo il punto di fuga individuato dall'analisi di tutti i frame a partire dal frame 0 (a). Al frame 435, il rumore accumulato nei frame precedenti porta ad un punto di fuga storico sbagliato, che però è già corretto al frame 469, quando l'informazione accumulata prevale sul rumore immagazzinato a causa di pendenze, curve e incroci.



## Capitolo 3

# Rilevazione delle corsie stradali

La percezione della strada, con la rilevazione delle corsie stradali (*lane detection*), era una delle capacità indispensabili di cui il veicolo TerraMax2 doveva essere dotato per il rispetto delle regole stradali. In questo capitolo sarà presentata una panoramica dell'algoritmo implementato per svolgere questo compito.

Un'osservazione subito necessaria riguarda i tempi di elaborazione: essendo presente sullo stesso elaboratore anche l'algoritmo di rilevazione degli ostacoli, l'algoritmo di rilevazione delle corsie deve avere un costo computazionale estremamente contenuto.

### 3.1 Immagine del giallo

Mentre le righe bianche sono analogamente utilizzate per tracciare i bordi della carreggiata e dividere le corsie di stesso senso di marcia, la segnaletica orizzontale statunitense si differenzia da quella italiana perché utilizza linee di colore giallo per dividere corsie con traffico in direzioni opposte. Questo ha reso perciò necessario lo sviluppo di un algoritmo che fosse in grado di rilevare anche questo tipo di linee, in particolare sia singole sia doppie. Per le righe bianche invece i possibili casi erano di riga tratteggiata o singola continua.

Osservando l'immagine di figura 3.1 è possibile osservare che la versione in bianco

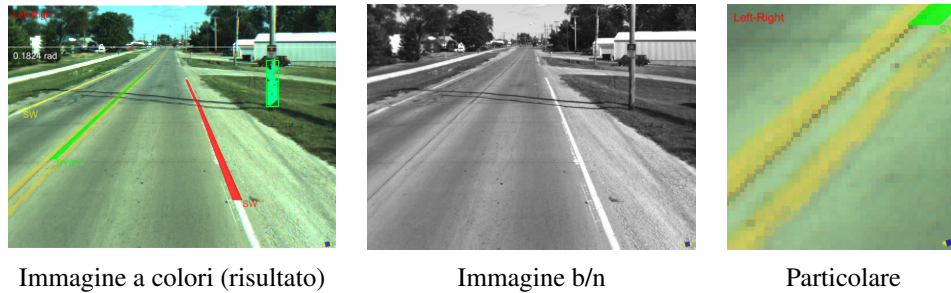


Figura 3.1: Studio su riga gialla.

e nero delle immagini non mette ha in sé per nulla evidenziate le righe gialle; è anche possibile constatare, esaminando il particolare ingrandito, che in questo caso la colorazione dell'asfalto ha una tonalità diversa dal grigio.

Andando a guardare i valori delle terne (R, G, B) possono essere presi due valori di esempio nell'immagine, trovando per esempio sull'asfalto (149, 186, 136) e sulla riga (200, 200, 102). Per evidenziare questa caratteristica si è costruita un'immagine ricavata dal rapporto R/B, che potesse mettere in rilievo appunto il maggior contenuto di rosso e minore di blu all'interno delle righe gialle, creando quella che nel resto questa trattazione essa sarà detta *immagine del giallo* (fig. 3.2).

Come è possibile osservare, la zona d'erba ai lati dell'immagine risulta essere più luminosa rispetto all'asfalto. In generale quindi non è conveniente applicare una soglia di binarizzazione su questa immagine per decidere cosa sia giallo nell'immagine o meno. Visto che l'immagine del giallo ha in sé evidenziate le righe gialle così come le righe bianche sono evidenziate nelle immagini in bianco e nero, l'elaborazione è suddivisa in due parti esattamente analoghe:

- nell'immagine in bianco e nero sono estratte le righe bianche;
- nell'immagine del giallo sono estratte le righe gialle.



Figura 3.2: Immagine del giallo.

## 3.2 Elaborazione di basso livello

### 3.2.1 Calcolo dell'IPM e estrazione dei pattern DLD

Le linee stradali causano in generale nell'immagine una caratteristica (*pattern*) evidente molto utilizzata in letteratura, detta pattern Dark-Light-Dark [11], ovvero scuro-chiaro-scuro, lungo le righe dell'immagine. L'ampiezza di questa caratteristica è però variabile in base all'ampiezza con la quale la riga è stata tracciata, e soprattutto, nell'immagine, in dipendenza dalla prospettiva: un tratto di riga lontana nell'immagine ha una larghezza inferiore rispetto ad un tratto di riga vicina. Per gestire in modo semplice questo problema, è possibile rimuovere l'effetto di prospettiva con una trasformazione di Inverse Perspective Mapping (IPM) [12], che è possibile conoscendo i parametri di calibrazione. L'IPM fornisce una vista dall'alto degli elementi appartenenti al piano stradale (fig. 3.3b).

Questa trasformazione si basa sull'ipotesi di terreno piatto, che per gli ambienti dove si sarebbe svolta la Urban Challenge era un'approssimazione accettabile. Per una corretta trasformazione prospettica è inoltre necessario conoscere l'angolo di pitch



(a) Immagine risultato in esame

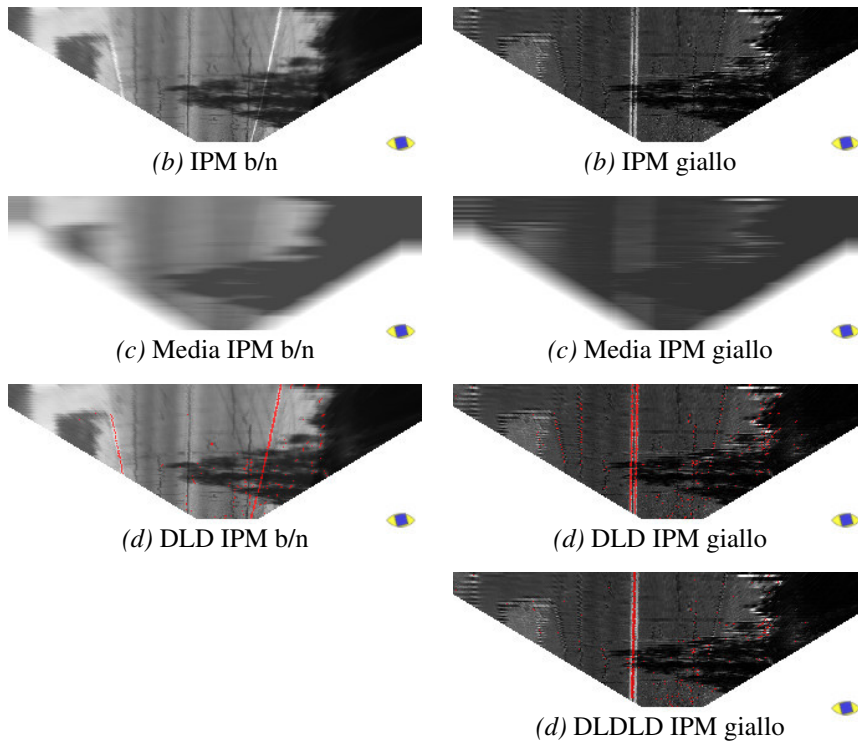


Figura 3.3: Elaborazione di basso livello

---

all'istante di acquisizione: grazie all'elaborazione presentata in appendice A questa informazione è disponibile; l'unico costo pagato in questo caso deriva dal fatto che la tabella di trasformazione IPM deve essere calcolata frame per frame, in quanto appunto uno degli angoli di calibrazione è variabile. Per questa ragione, il calcolo delle tabelle di trasformazione, che in passato non aveva richiesto particolari accorgimenti perché doveva essere effettuato solo una volta all'inizio dell'elaborazione, è stato ottimizzato limitandone l'onere computazionale.

La trasformazione di rimozione prospettica non richiede di portarsi in un dominio in cui ogni pixel rappresenti regioni del mondo reale quadrate: al contrario, lo studio e le sperimentazioni effettuate hanno portato a concludere che fosse più efficace una trasformazione che mantenesse più informazione possibile a livello laterale, in modo da poter meglio analizzare i pattern DLD. A livello frontale rispetto al veicolo, al contrario, una risoluzione elevata non porta a particolari benefici, in quanto rischia di aggiungere informazione fittizia (per esempio espandendo su molti pixel regioni del mondo lontane che nell'immagine appaiono con pochi pixel) non utile all'ottenimento dei risultati, peraltro aumentando le dimensioni dell'immagine con un'impennata dei costi computazionali.

Il primo passo dell'estrazione dei pattern DLD per la ricerca delle linee singole avviene semplicemente calcolando per ogni pixel dell'IPM la differenza con due pixel laterali ad una certa distanza fissa (valore DLD). Per la ricerca delle linee doppie, nell'immagine del giallo è ricercato anche un valore per il pattern di tipo DLDDL, in modo analogo, ottenendo un'altra immagine che subisce gli stessi processi. Da osservare è che non è corretto presumere che i due livelli scuri del pattern DLD debbano essere analoghi: se si osserva la figura 3.4 si può per esempio osservare come in questo caso la linea bianca separi due regioni a tono di grigio molto diverso.

I valori trovati sono quindi soglia e binarizzati per stabilire quali sono i pattern candidati ad appartenere ad una riga (fig. 3.3d, in rosso sovrapposte alle immagini che le hanno originate). Considerato però che le condizioni di illuminazione di una riga possono essere molteplici, la soglia con cui è confrontato il valore DLD è adattata al valore di luminosità presente nella regione dell'immagine IPM nell'intorno del pixel



Figura 3.4: Esempio di linea (bianca sulla destra) che separa due regioni a differente tono di grigio.

di elaborazione. Per conoscere il valore di luminosità locale, l'algoritmo utilizza la media dei valori in un intorno di larghezza fissa (solitamente un numero potenza di 2 per ragioni computazionali) e altezza uno. La media è calcolata preventivamente su tutta l'immagine in tempi rapidissimi utilizzando per ogni pixel i risultati parziali del pixel alla sua sinistra (fig. 3.3c). Per evitare però che questo tipo di soglia di vantaggi troppo le zone d'ombra dell'immagine, ai pixel dell'immagine della media di valore inferiore di un certo valore minimo stabilito è assegnato questo valore minimo stesso. In questo modo è possibile rilevare anche linee stradali poste in ombra, restando però più restrittivi nelle zone dove proprio per scarsa illuminazione, il contenuto d'informazione è minore e quindi meno affidabile.

### 3.2.2 Mascheratura degli ostacoli

La trasformazione IPM fornisce una corretta visione dall'alto di ciò che appartiene al piano stradale. Oggetti come automobili o pali sono invece deformati e possono assumere nell'immagine IPM una forma simile ad una linea. Tuttavia, utilizzando i risultati del rilevatore di ostacoli, è possibile mascherare appunto gli ostacoli rilevati dalle immagini e non considerare le regioni dell'immagine da essi occupati durante la ricerca delle linee. Riprendendo l'esempio di figura 3.1, l'algoritmo calcola

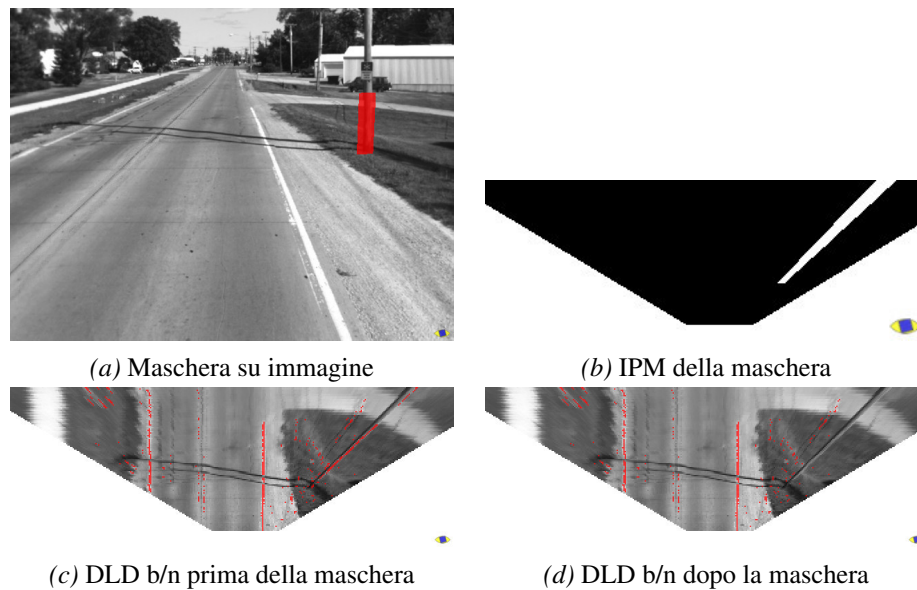


Figura 3.5: Mascheramento degli ostacoli di figura 3.1. Il pattern DLD causato dal palo è cancellato.

per l'ostacolo presente una regione poligonale che lo ricopre e crea un'immagine di mascheratura disegnandovi sopra il poligono creato. Nella figura 3.5a, l'immagine maschera in rosso è sovrapposta all'immagine in bianco e nero. L'immagine di maschera è quindi trasformata nel dominio IPM (fig. 3.5b), in modo da essere utilizzata per spegnere tutti i pixel delle immagini DLD e DLDDL nelle posizioni corrispondenti. Occorre osservare che non è possibile intervenire con la mascheratura disegnando i poligoni (con qualsiasi tono di grigio o colore) nelle immagini da elaborare, in quanto si creerebbero dei pattern DLD fittizi.

### 3.2.3 Etichettatura (Labeling)

Nelle immagini binarizzate LHL e LHLHL sono quindi ricercate e individuate le regioni continue, a cui è assegnata un'etichetta (*label*). Nella visualizzazione di fi-

gura 3.6, che riprende l'esempio di figura 3.3, ogni colore rappresenta una diversa etichetta assegnata. Regioni etichettate troppo piccole sono subito scartate.

La regola di espansione delle regioni etichettate (in breve *etichette*) permette di collegare due pixel separati verticalmente da un pixel spento; in questo modo è possibile mantenere un'unica etichetta anche per linee stradali parzialmente deteriorate, effetto tra l'altro favorito dalla scelta di utilizzare IPM con poche righe: eventuali tratti corrotti di altezza inferiore a due pixel sono corretti già a questo livello.

### 3.2.4 Rimozione dal b/n dei label del giallo

Una linea gialla in buono stato può avere una luminosità significativa anche nell'immagine in bianco e nero. Per evitare di processare la stessa riga due volte con il rischio al termine dell'elaborazione di non saper stabilire se la linea individuata sia di colore bianco o giallo, i pixel che superano con successo la fase di etichettatura nell'immagine DLD sul giallo sono rimossi dall'immagine DLD sul b/n, prima che questa subisca l'operazione di etichettatura.

### 3.2.5 Conversione a coordinate mondo

Le etichette sulle immagini IPM costituiscono le candidate ad essere considerate pezzi di linea stradale. La loro forma in questo momento dell'elaborazione è però ancora influenzata dal fatto che i pixel della IPM non rappresentino una regione quadrata del mondo; oltre a causare problemi di stima della lunghezza, l'effetto più evidente riguarda gli angoli, che sono ruotati in modo anomalo. Per ovviare a questo problema, a questo punto dell'elaborazione tutti i punti appartenenti alle etichette sono convertiti in coordinate mondo, cosicché tutte le lunghezze e gli angoli che saranno calcolati da questo momento saranno coerenti con la realtà.

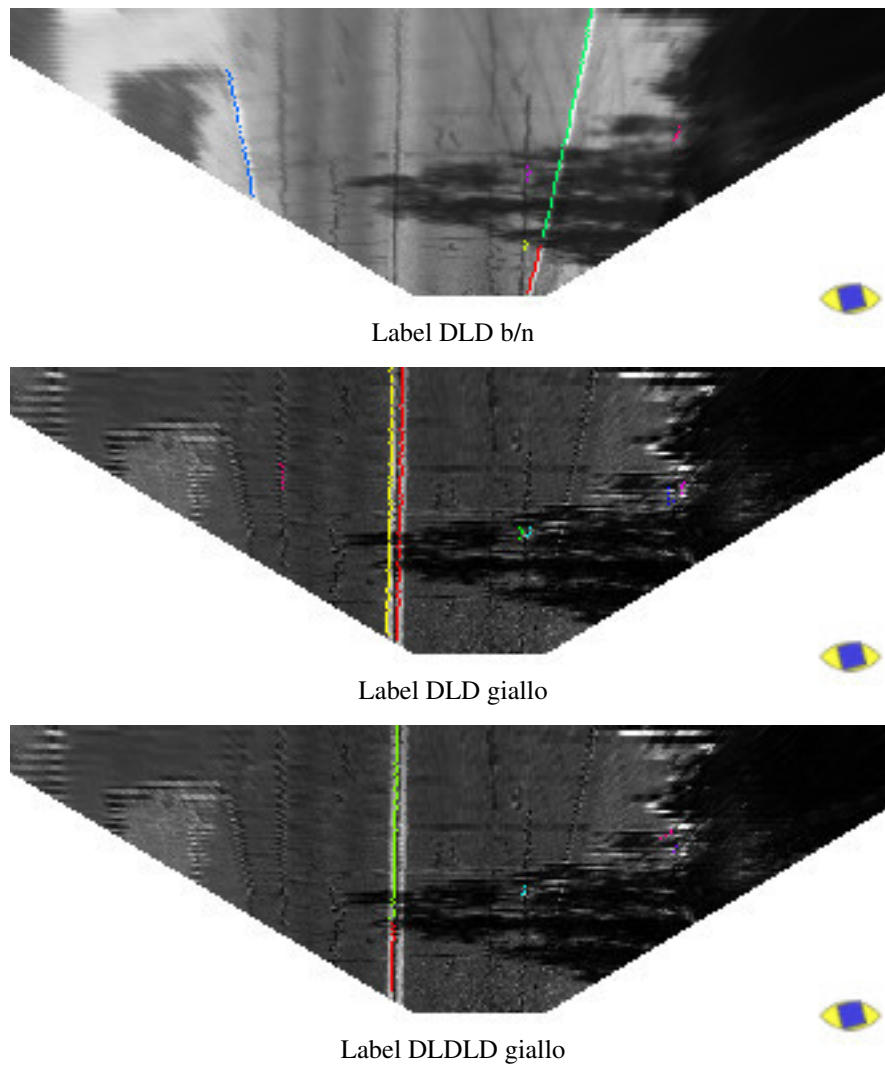


Figura 3.6: Risultato del labeling delle immagini di figura 3.3d.

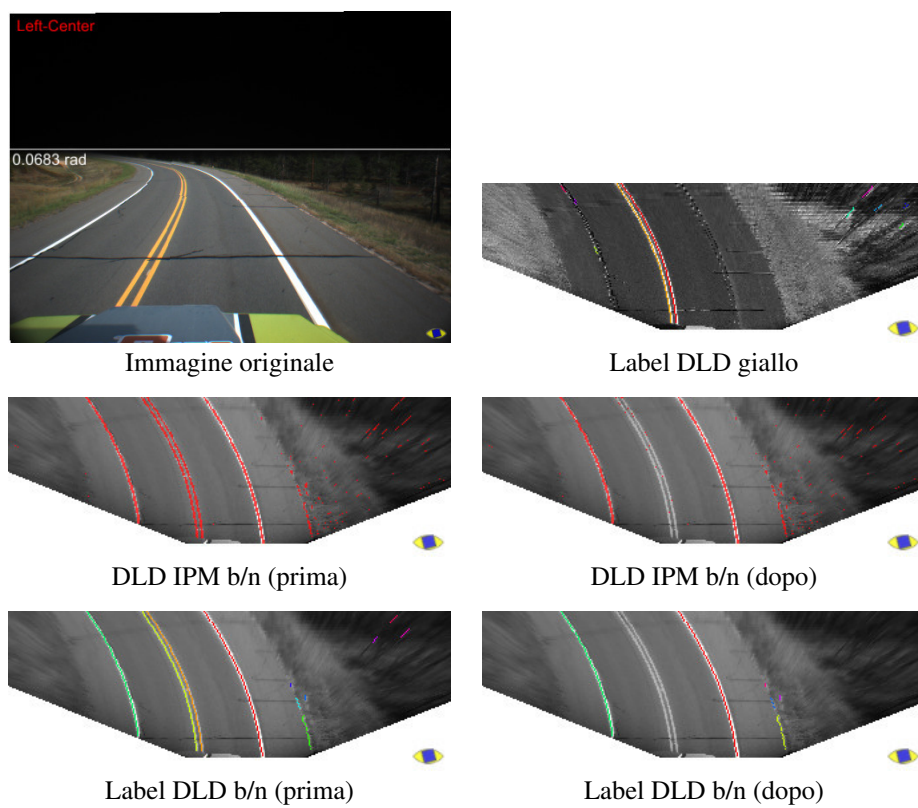


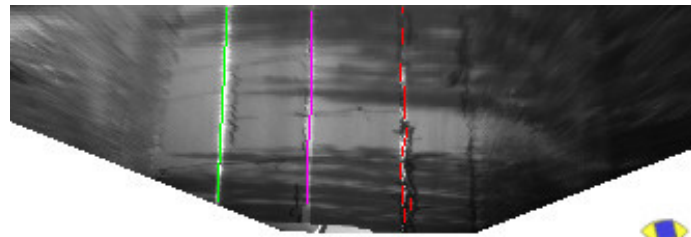
Figura 3.7: Rimozione delle etichette gialle dalle immagini b/n.

## 3.3 Creazione del modello stradale

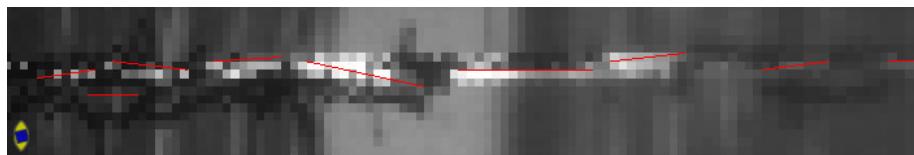
### 3.3.1 Individuazione della linea madre e creazione delle linee spezzate

Rilevare le corsie in modo generale non è un problema semplice. Possono essere applicate alcune semplificazioni, ma non tutte hanno applicazione reale. Per esempio, come ha mostrato l'esempio di figura 3.3, non è vero che le righe stradali sono sempre parallele fra di loro. L'algoritmo procede ricercando tra le etichette la linea (*linea madre* che con più confidenza può essere considerata linea stradale, e in base ad essa classifica le altre etichette presenti come possibili pezzi di una linea a sinistra, a destra, o a sinistra o destra distanti la larghezza di due corsie (in questo caso si dirà *sinistra-sinistra* e *destra-destra*). La scelta della linea madre avviene considerando nelle varie immagini LHL e LHLHL la linea che accumuli più punteggio, considerando un peso maggiore per i punti del mondo più vicini al veicolo: questo perché questi punti possono essere con meno probabilità un ostacolo deformato nella proiezione IPM. Se non è individuata nessuna linea madre di sufficiente punteggio, la ricerca delle linee per il frame corrente è considerata fallita.

Nella figura 3.8 è mostrato un primo esempio di scelta di linea madre (disegnata in viola nell'immagine di classificazione delle linee), con classificazione delle altre etichette a possibili parti di linee adiacenti. Per quanto riguarda l'insieme delle linee rosse, ovvero quelle classificate come poste ad una distanza di una corsia dalla linea madre sulla sua destra, è possibile osservare nell'esempio come in caso di linee parzialmente danneggiate i pixel agli estremi delle etichette non possano essere considerati affidabili: in questi casi può infatti capitare che il filtro DLD metta in rilievo pixel che con la linea in realtà non c'entrano. Per questa ragione nella fase di prolungamento della linea, ogni volta che un segmento è aggiunto, la linea subisce un'operazione di piccolo accorciamento per evitare che l'ultimo pezzo introdotto dia una direzione priva di senso, che impedirebbe l'inclusione degli altri tratti di linea. La figura risultato mostra come in questo caso la linea sia ricostruita correttamente. Un altro esempio di questo processo di categorizzazione e unione di etichette è dato dal caso rappresentato nelle figure 3.9, 3.10, e 3.11. In questo caso la linea madre è



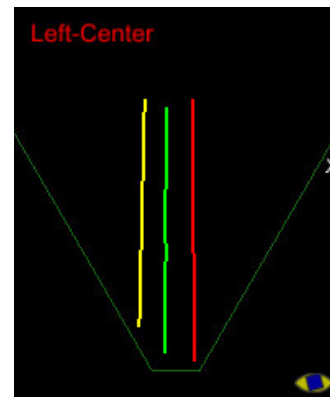
Classificazione linee



Tratti di linea, particolare (ruotato in senso orario)



Risultato



Vista dall'alto dei risultati

Figura 3.8: Classificazione di linee rispetto alla linea madre, e costruzione della spezzata per la riga destra. Nei risultati, viola=linea madre, rosso=destra, verde=sinistra, azzurro=destra-destra, giallo=sinistra-sinistra.

ottenuta da etichette separate e quindi unite, come mostra figura 3.10b.

Nelle figure 3.10c è possibile vedere come la linea madre classifichi nel dominio b/n una linea a sinistra-sinistra, mentre nel dominio del giallo un certo numero di etichette sono classificate come a sinistra (ingrandimento di figura 3.10d). Verificando quindi la compatibilità, queste linee sono unite in un'unica spezzata, come è possibile vedere nel risultato mostrato in figura 3.11.

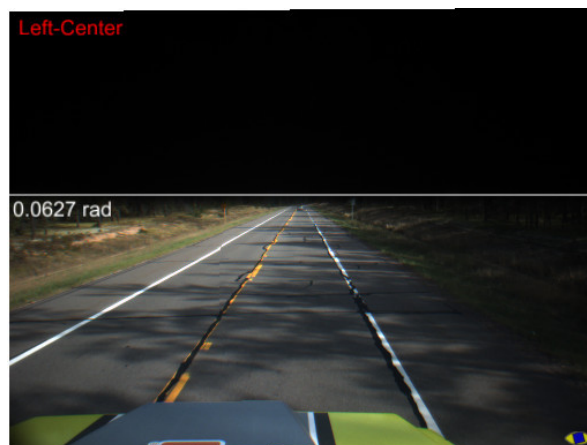


Figura 3.9: Immagine in ingresso.

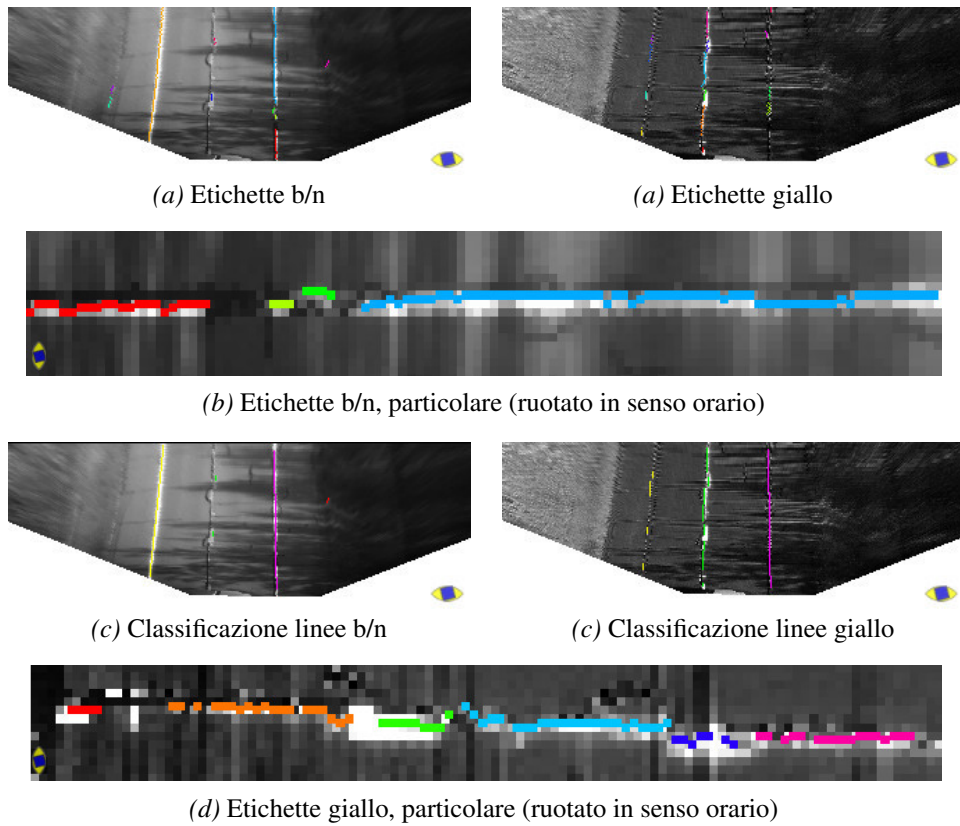


Figura 3.10: Ricostruzione di linea madre, classificazione e ricostruzione altre linee per l'esempio di figura 3.9. Nella classificazione, viola=linea madre, rosso=destra, verde=sinistra, azzurro=destra-destra, giallo=sinistra-sinistra.



Figura 3.11: Risultato. Rosso=destra, verde=sinistra, azzurro=destra-destra, giallo=sinistra-sinistra.

### 3.3.2 Segmentazione

Etichette eventualmente curve sono approssimate con una spezzata come mostrato in figura 3.12: è fissata una soglia in metri, e se la linea retta che unisce i due estremi dell'etichetta passa più lontano di questa soglia da uno dei punti allora l'etichetta è spezzata in due, aggiungendo un punto appartenente all'etichetta in corrispondenza del punto più lontano. L'operazione è ripetuta iterativamente fino a quando la linea ottenuta rientra nelle specifiche determinate dal parametro settato.

### 3.3.3 Curve

E' interessante osservare come avviene la categorizzazione delle etichette da parte della linea madre in caso di linea curva. Prima di procedere con la categorizzazione, la linea madre subisce l'operazione di segmentazione: in questo modo è possibile conoscere la curvatura/inclinazione attesa per le altre linee a varie distanze. Le altre etichette sono accettate se la loro inclinazione (data in primissima approssimazio-

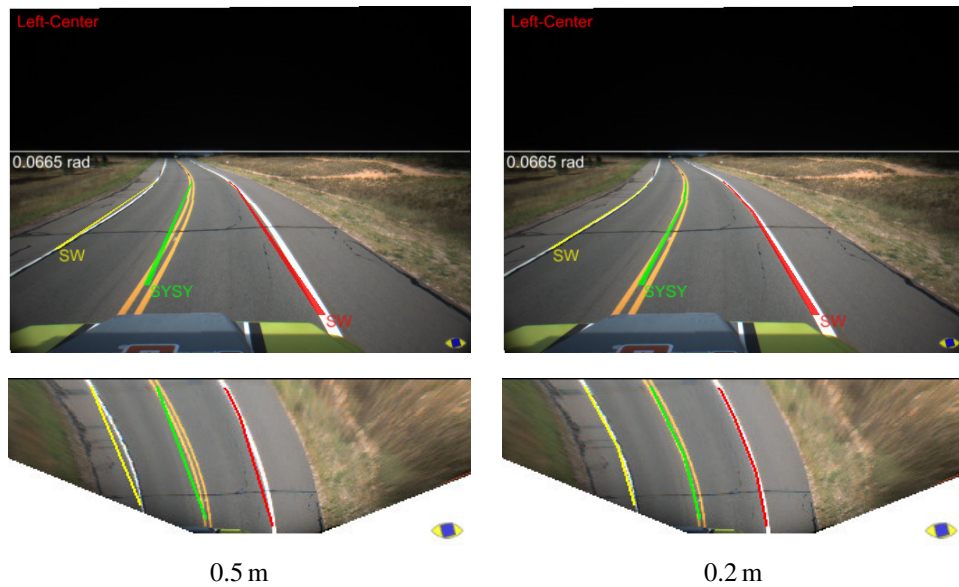


Figura 3.12: Uso di due diverse soglie di segmentazione.

ne dalla retta unente il primo e l'ultimo punto) all'altezza del proprio baricentro è compatibile con quella della linea madre.

Nell'esempio di figura 3.13, i due pattern DLD creati dal marciapiede alla destra della linea madre sono entrambi considerati compatibili come distanza e inclinazione, e uno di loro è infatti scelto come linea destra in uscita.

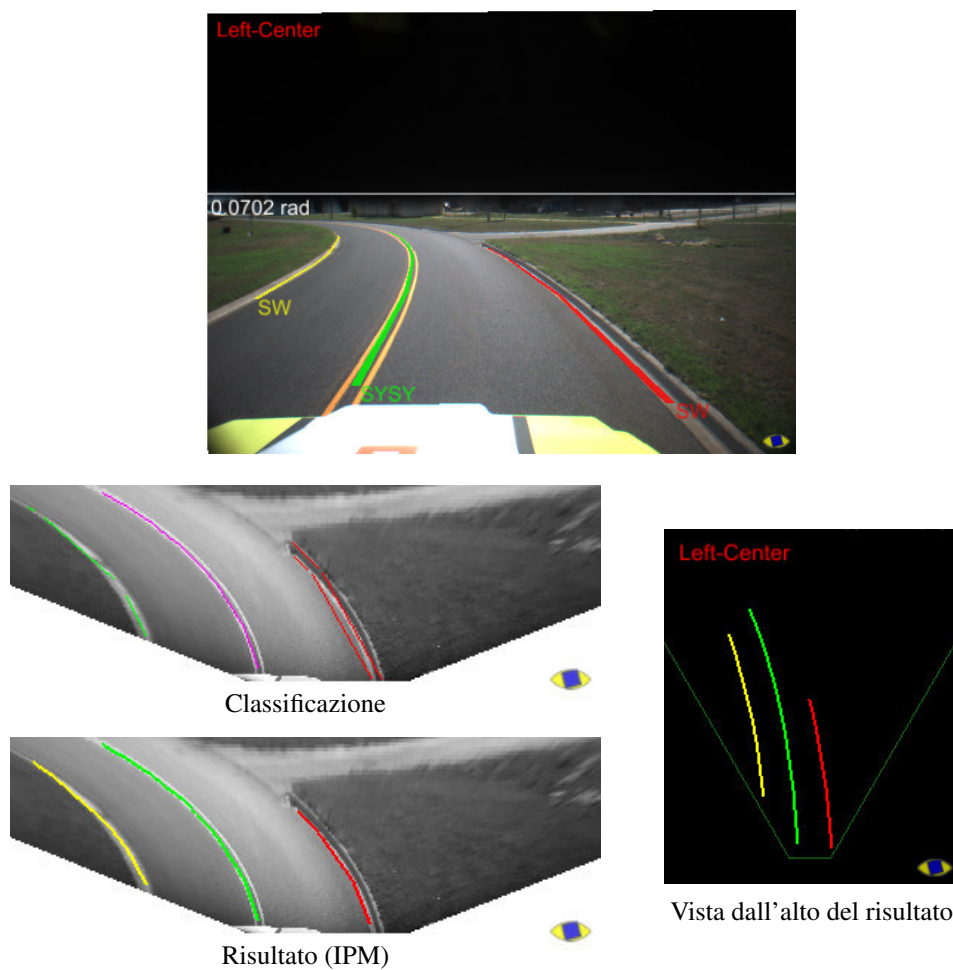


Figura 3.13: Classificazione di label in funzione della linea madre in caso di strada curva. Nei risultati, viola=linea madre, rosso=destra, verde=sinistra, azzurro=destra-destra, giallo=sinistra-sinistra.

### 3.3.4 Linee tratteggiate

Nel caso di linee tratteggiate, la ricerca delle linee riserva un numero maggiore di ambiguità, in quanto numerosi aspetti del mondo reale possono creare un pattern DLD compatibile con un tratteggio. Inoltre è necessario osservare che molto spesso i tratteggi delle linee discontinue sono orientati in modo irregolare rispetto al senso di marcia, e non sempre quindi un tratteggio è orientato per puntare al tratteggio successivo. Perciò per riuscire ad individuare questo tipo di linee occorre non essere troppo stringenti sui parametri di inclinazione relativa tra i tratteggi. Per queste ragioni da subito la ricerca delle linee tratteggiate è stata introdotta con una fase di tracking temporale: una linea di questo tipo può essere validata solo se è riconosciuta per 3 frame consecutivi.

La ricerca della linea avviene innanzitutto categorizzando nell'immagine quelli che sono i segmenti che per la loro lunghezza sono compatibili con un possibile tratteggio (fig. 3.14b). Un tratteggio "seme" è quindi scelto, nell'intorno di una linea tratteggiata precedentemente trovata o in un'area vicina del mondo, e da questo è fatta una valutazione degli altri tratteggi validati per creare da esso una nuova linea (fig. 3.14c).

Come è possibile vedere dalla figura 3.14d, la linea spezzata rossa individuata al frame corrente si trova ad essere vicini con le spezzate verde e blu dei due frame precedenti, ed è pertanto validata e mandata in uscita (fig. 3.14e).

Resta da osservare come l'algoritmo di ricerca per le linee tratteggiate sia stato sviluppato tenendo in considerazione le condizioni che si sarebbero potute trovare alla Urban Challenge: in particolare era noto che non ci sarebbe potuta essere stata più di una linea tratteggiata, e i tratteggi sarebbero stati piuttosto distanti fra di loro per cui tre tratteggi sarebbero stati sufficienti a coprire la distanza di 40 metri che si intendeva osservare. Queste limitazioni (una sola linea tratteggiata rilevabile, composta al massimo di tre tratteggi) dovranno essere negli sviluppi futuri rimosse.

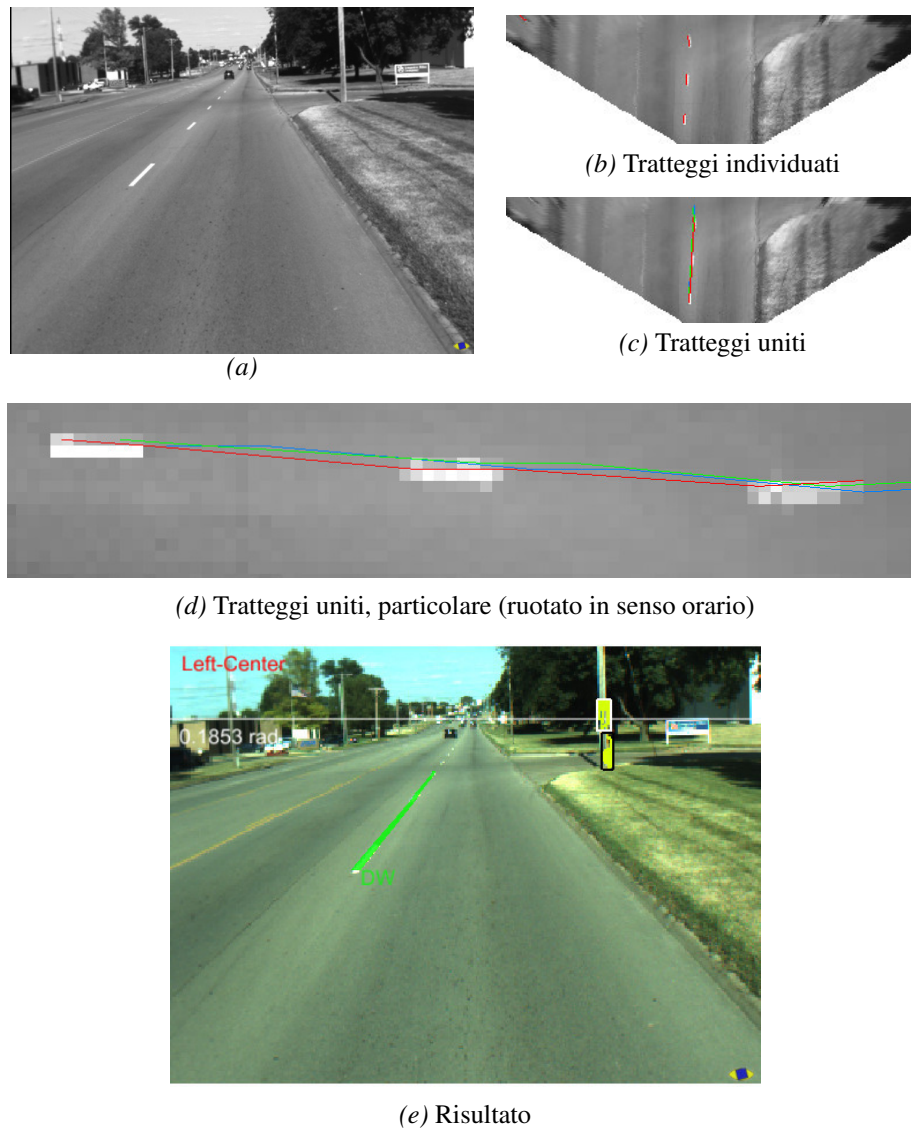


Figura 3.14: Ricerca e unione di tratteggi. Nel particolare ingrandito, la spezzata rossa indica la linea rilevata al frame corrente, la linea verde al frame precedente e la linea blu al frame precedente ancora. Le spezzate sono coerenti e la linea è pertanto validata.

### 3.3.5 Tipi di linea

Le linee sono mandate in output insieme ad un dato che ne identifica il tipo:

- Doppia gialla continua (SYSY, solidYellow-solidYellow)
- Gialla continua (SY, solidYellow)
- Gialla discontinua (DY, dashedYellow)
- Bianca continua (SW, solidWhite)
- Bianca discontinua (DW, dashedWhite)

### 3.3.6 Guardrail

L'individuazione dei pattern di basso livello che possono costituire una linea è messa in crisi, oltre che dagli oggetti verticali proiettati che però possono essere abbastanza facilmente individuati dall'algoritmo di rilevazione degli ostacoli e quindi rimossi, da pattern longitudinali alla strada che possono essere creati per esempio da strutture come i guardrail (fig. 3.15b), o le loro ombre.

Per questa ragione l'algoritmo ricerca in un primo momento la linea madre solo in una regione lateralmente vicina (una linea cioè che presumibilmente delimita la corsia in cui il veicolo si trovi) e solo in caso di mancata individuazione estende la ricerca anche più lontano. Ciò nonostante, come mostra l'esempio di figura 3.15, nell'immagine IPM il guardrail può generare un pattern DLD ad una distanza perfettamente compatibile con quella di una linea sinistra-sinistra, ed è pertanto validata in uscita.

Questo tipo di problema non è stato affrontato per la Urban Challenge, in quanto i membri della squadra addetti al design del pianificatore del moto avevano già deciso da tempo di trascurare le informazioni riguardo alle corsie adiacenti (anche per quanto riguardava le operazioni di sorpasso), prendendo in considerazione solo i dati relativi alla corsia in cui il veicolo si trovava. Una possibile risoluzione al problema potrebbe venire dal metodo proposto in [13].

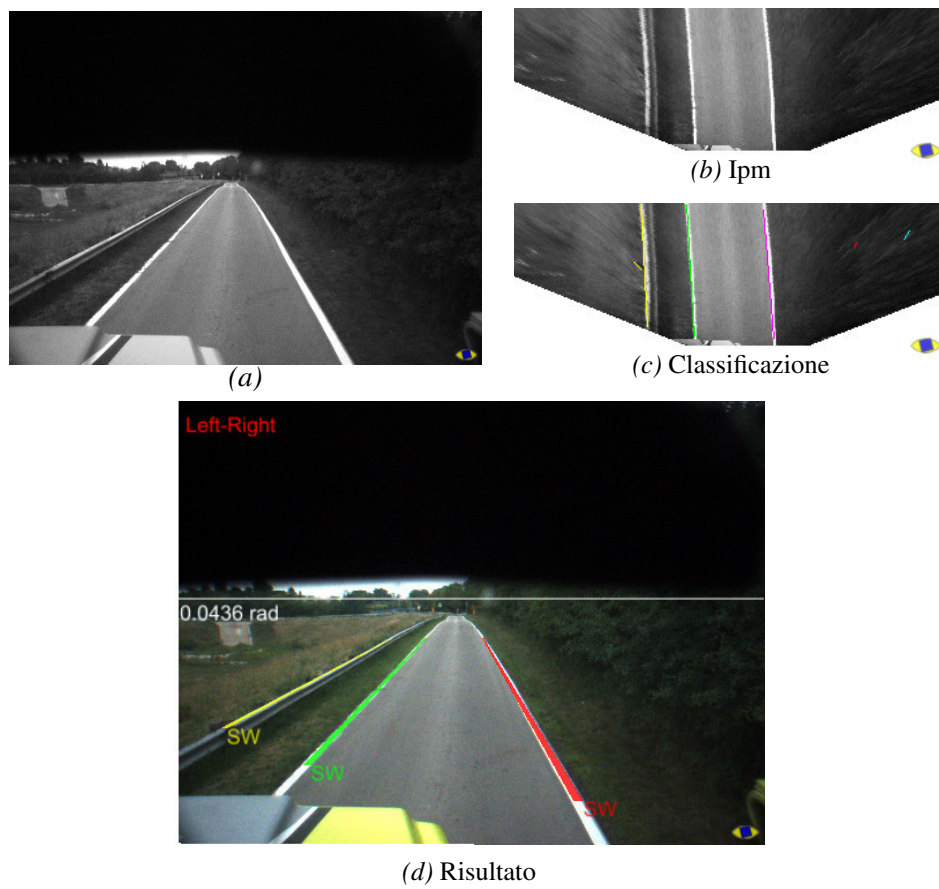
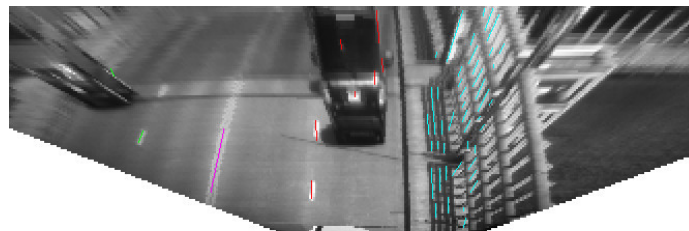


Figura 3.15: Il guardrail è rilevato come pattern DLD, è classificato come linea a distanza di due corsie dalla linea madre e quindi erroneamente validato. Nei risultati, viola=linea madre, rosso=destra, verde=sinistra, azzurro=destra-destra, giallo=sinistra-sinistra.

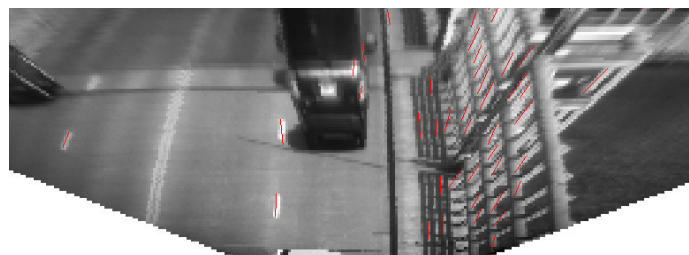
Un altro esempio piuttosto indicativo dei problemi che possono essere dati dalla presenza di una barriera parallela alla strada è visibile in figura 3.16: essa infatti lascia passare la luce attraverso le sue aperture creando pattern compatibili ad una caratteristica DLD, che in questo caso è validata come linea. Inoltre la barriera genera nell'immagine IPM un gran numero di pattern compatibili con un tratteggio di una linea tratteggiata: questo rumore impedisce di rilevare correttamente la linea tratteggiata realmente presente.



Risultato



Linee compatibili con la linea madre



Tratteggi candidati

Figura 3.16: L'ombra della barriera produce un'ombra con pattern DLD, ed è classificata come linea. Nell'immagine dei tratteggi sono presenti molti elementi di disturbo, che impediscono di rilevare correttamente la linea tratteggiata.

### 3.4 Tracking

Allo scopo di irrobustire i risultati e permettere al modello della strada, che nel nostro caso ha come unico parametro fissato il range di possibili larghezze delle corsie, di deformarsi seguendo la conformazione della strada, è stata aggiunta un'operazione di tracking delle linee.

Oltre alla ricerca normale, per l'implementazione del tracking attorno alle linee trovate nel frame precedente è individuata una regione dove sono ricercate etichette eventualmente presenti. Con queste etichette individuate, l'algoritmo cerca di creare una nuova linea spezzata, che poi prolunga ulteriormente se possibile utilizzando come candidate tutte le etichette presenti nell'immagine. Alla linea spezzata individuata in questo modo è assegnato un valore maggiorato attraverso un fattore correttivo, che le consente una maggiore probabilità di essere scelta come linea madre o di essere validata.

Il sistema implementato permette al modello di strada di deformarsi seguendo i cambiamenti della strada, come nell'esempio di figura 3.17 in cui la comparsa di una nuova corsia è gestita in modo eccellente.

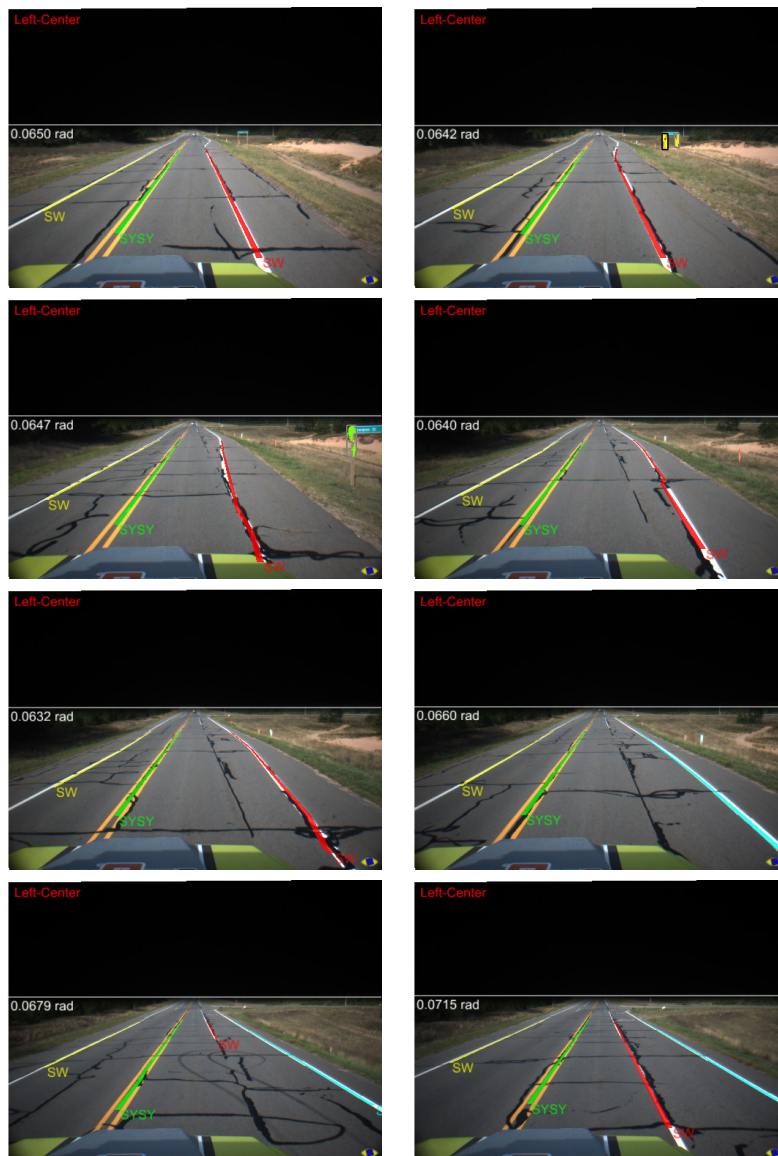


Figura 3.17: Tracking. Nei risultati, viola=linea madre, rosso=destra, verde=sinistra, azzurro=destra-destra, giallo=sinistra-sinistra.

### 3.5 Conclusioni e sviluppi futuri

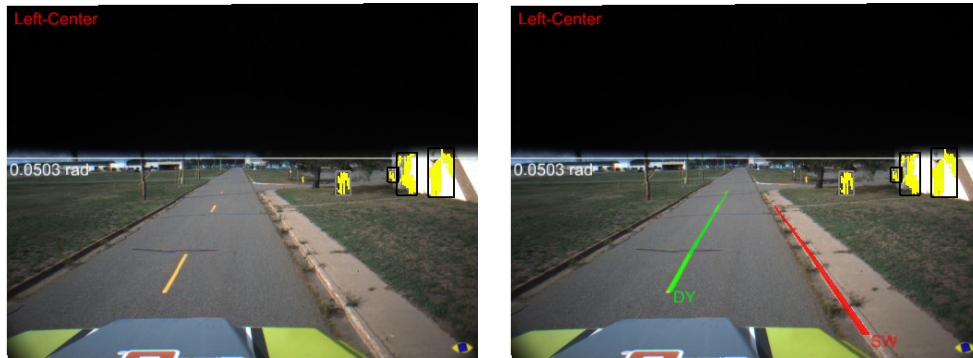


Figura 3.18: Rilevazione di una linea tratteggiata gialla e del bordo di un marciapiede.

Nell'esempio di figura 3.18, è mostrato un esempio di funzionamento in condizioni particolari, con la presenza di un'unica linea tratteggiata gialla e dei bordi del marciapiede. Molti dei test sul sistema sono stati effettuati su strade di questo tipo, nell'area della base aerea dismessa di Marquette (MI, Usa).

L'algoritmo di lane detection presentato riesce ad individuare con successo differenti tipologie di linee nelle immagini. Grazie al tracking implementato, è possibile adattare il modello della strada mentre questo cambia, ad esempio nel caso di innesto di una nuova corsia. I tempi di computazione riscontrati su un sistema a processore P4@2.8GHz sono stati di 10 ms. Per un impiego effettivo in un sistema di assistenza alla guida l'algoritmo necessita però di essere migliorato per quanto concerne la rilevazione delle righe tratteggiate e la robustezza a pattern come guardrail o ombre di guardrail. Inoltre, in un possibile impiego su immagini monoscopiche, in assenza quindi delle informazioni provenienti dai moduli stereo, deve essere inserito un processo di stabilizzazione e una serie di controlli per accertare che la linea individuata non sia costituita in realtà da un ostacolo.

Un'importante passo avanti per l'algoritmo sarebbe inoltre il poter individuare la strada indipendentemente dalle righe, ma semplicemente in base ad un'analisi sul

colore. Questo risultato non è però facilmente conseguibile, a causa ad esempio degli effetti dovuti alle ombre, che impediscono l'individuazione di un vero contrasto tra superficie stradale e aree circostanti.



## Capitolo 4

# Determinazione del profilo del terreno

La presenza di pendenze della strada introduce un grado di complessità aggiuntivo per i sistemi di percezione per autoveicoli. Per esempio, un algoritmo di rivelazione ostacoli basato su visione stereo (si veda app. B) può essere indotto a considerare un cambio di pendenza come ostacolo, e lo stesso può accadere ad un sistema basato sui dati provenienti da radar o laserscanner. Inoltre, tutti gli algoritmi che si basano sull'assunzione di terreno piatto in questo caso giungono a risultati incorretti. Per esempio, una trasformazione di rimozione della prospettiva (IPM), come quella utilizzata dall'algoritmo di ricerca delle corsie stradali (cap. 3), in queste condizioni genera una visione deformata del mondo reale, come quella visibile in figura 4.1, dove la pendenza causa l'impressione di una strada che improvvisamente si restringa. L'estrazione del profilo del terreno costituirebbe quindi un'utile informazione per correggere questi effetti.

In [14, 15], gli autori affrontano il problema assumendo piccoli cambi di pendenza e di avere una disperazione una DSI (Disparity Space Image) con pochi contributi di rumore, in particolare che non restituisca punti nel mondo di coordinata sotto il livello del terreno. Questo approccio si è dimostrato efficace in ambienti strutturati,

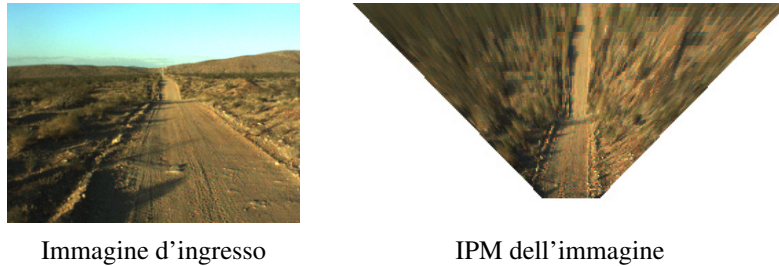


Figura 4.1: IPM di un terreno pendente.

ma fallisce in ambito off-road dove entrambe le semplificazioni (cambi di pendenza contenuti e buona ricostruzione stereo) non trovano riscontro. In questo studio è preso in esame un metodo che ha come obiettivo la percezione del profilo del terreno anche in ambiente off-road, senza vincoli sulle massime pendenze attese.

## 4.1 Estrazione del profilo del terreno

Come anticipato nel paragrafo 1.2, la linea di correlazione del terreno presente nella V-Disparity Image contiene informazioni sulla posizione e forma (profilo delle pendenze) del terreno. Lo scopo dell'algoritmo che si vuole realizzare è quindi individuare la linea di correlazione nella V-Disparity Image, dalla quale poi, attraverso semplici calcoli geometrici, è possibile risalire al profilo del terreno.

All'interno della mia tesi di laurea [7], riprendendo lo studio di T.A. Williamson [16], ho già avuto modo di trattare l'operazione di *warping* (distorsione) un'immagine, che consiste nel deformare l'immagine di sinistra facendo scorrere le sue righe lungo l'asse  $u$  di una quantità pari alla disparità del terreno corrispondente all'altezza  $v$  nell'immagine (questa informazione è disponibile dalla retta di correlazione, si veda l'app. A). L'effetto di questa operazione è visibile in figura 4.2 dove l'immagine ottenuta (detta **warped image**, immagine distorta) sembra corrispondere perfettamente all'immagine di destra (questo è vero però solo per gli elementi del terreno).

L'applicazione dell'operazione di *warping* sulla V-Disparity Image (con ottenimento

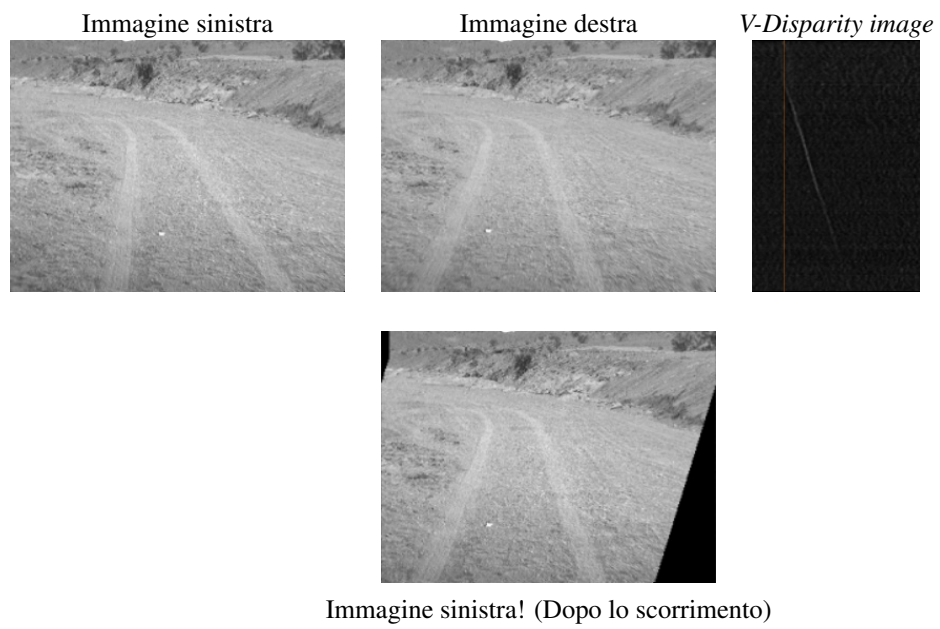


Figura 4.2: Effetto dello scorrimento dell'immagine sinistra secondo i valori dettati dalla retta di correlazione.

della *Warped VDIImage*), consente di semplificare la rivelazione dei punti appartenenti alla linea di correlazione, in quanto essi vanno a concentrarsi in una fascia verticale dell'immagine (fig. 4.3)

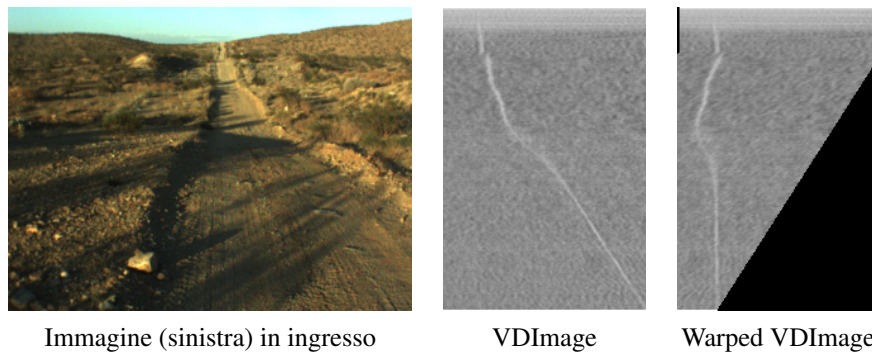


Figura 4.3: Effetto dello scorrimento dell'immagine sinistra secondo i valori dettati dalla retta di correlazione.

Dopo questo passo, l'elaborazione si concentra sull'estrazione dei massimi per riga nella *Warped VDIImage* 4.4; la posizione dei massimi è calcolata con precisione sub-pixel, interpolando mediante parabola i valori nell'intorno del massimo (trovato nel dominio discreto); come sarà possibile vedere, questo porta ad un miglioramento dei risultati, con l'estrazione di un profilo più continuo.

A causa del rumore presente nella *V-Disparity Image*, i massimi individuati devono essere sottoposti ad un'operazione di filtraggio onde ottenere i soli punti realmente appartenenti alla linea di correlazione, in particolare eliminando i massimi isolati. Essendo i punti di massima correlazione situati lungo una linea quasi verticale, si è utilizzato un filtro che pesasse maggiormente i punti in tale posizione. A questo proposito è stato realizzato un filtro per la ricerca del vicinato con una forma rettangolare molto restrittivo in senso orizzontale e permissivo in verticale. Questo filtro dà risultati migliori di un semplice filtro che analizzi un intorno circolare. I massimi ottenuti sono mostrati in figura 4.5 in cui è ora rimosso l'effetto di warping (operazione di warping). In questo caso, i parametri del filtro impostati sono larghezza del

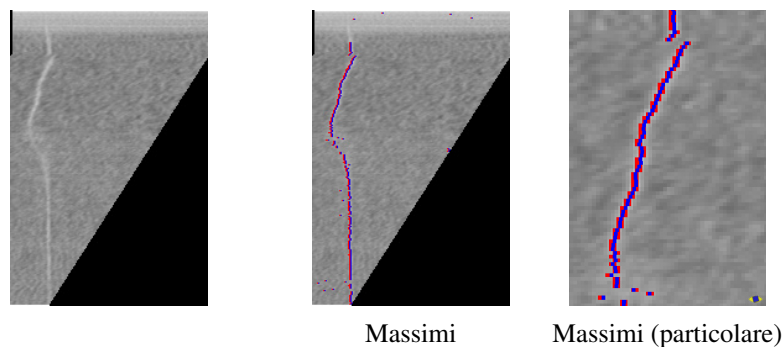


Figura 4.4: V-Disparity image warped e massimi trovati. In rosso sono visualizzati i punti di massima correlazione normali, mentre in blu quelli interpolati. Come si può già osservare, l'andamento dei punti blu è più continuo.

box rettangolare 2.2 pixel, altezza 9 pixel e percentuale di punti nel vicinato pari al 75%. Come è possibile osservare, il filtraggio risulta essere piuttosto severo, ma si è constatato che parametri meno restrittivi porterebbero alla validazione di punti di rumore, con conseguenze molto dannose sul profilo del terreno che si otterrebbe.

L'ultima operazione necessaria per tracciare il profilo del terreno consiste nell'eseguire una trasformazione da  $(v,d)$  coordinata nella V-Disparity image a coordinate mondo  $(x,y,z)$ , operazione che è possibile generando una coppia fittizia di punti da utilizzare per un'operazione di triangolazione:

- $(v,u+d)$  nell'immagine sinistra
- $(v,u)$  nell'immagine destra

dove  $u$ , colonna dell'immagine, è scelta arbitrariamente al centro dell'immagine, e infatti delle  $(x,y,z)$  in uscita la coordinata  $y$  è scartata, in quanto il profilo del terreno risiede sul piano  $(x,z)$ .

Il profilo è generato a partire dall'origine del sistema  $(x,z)$  (punto  $(0,0)$ ) e i punti calcolati al passo precedente sono inseriti serialmente, scartando i punti che, per approssimazione o rumore, abbiano una coordinata  $x$  inferiore ai punti già inseriti (la

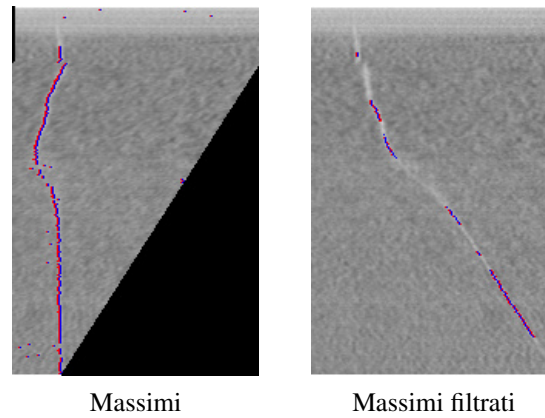


Figura 4.5: Massimi trovati nella V-Disparity image warped e massimi filtrati.

coordinata  $x$ , distanza frontale dal veicolo, deve essere sempre crescente). Il profilo è quindi costruito dalla spezzata passante per questi punti (fig. 4.6). Il profilo è generato sia impiegando i punti di massima correlazione normale (rossi), sia servendosi dei punti interpolati (blu) per poter osservare i miglioramenti ottenuti (fig. 4.7).

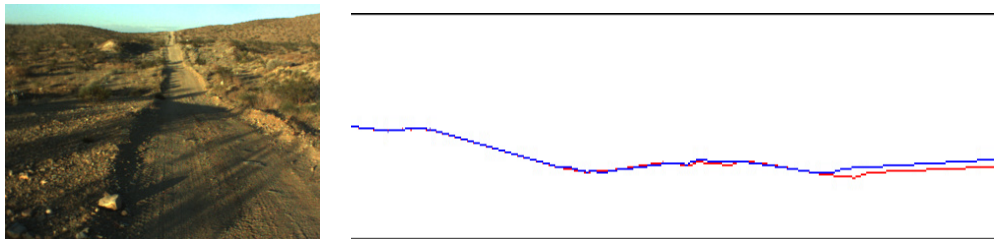


Figura 4.6: Immagine telecamera e profilo del terreno dove il tratto rosso indica il profilo ottenuto dai punti normali mentre quello blu è relativo al profilo ottenuto dai punti migliorati.

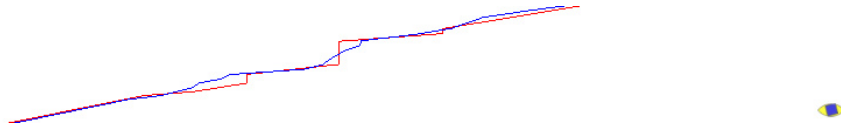


Figura 4.7: Particolare ingrandito di un esempio in cui l'interpolazione consente l'eliminazione dei gradini che si avrebbero utilizzando la disparità discreta.

## 4.2 Stirare la superficie stradale

In figura 4.8 è possibile vedere il risultato di questa operazione. La seconda parte di questo studio consiste nella realizzazione di un metodo per la rimozione della pendenza del terreno dalle immagini, ottenendo cioè le immagini che si avrebbero se la superficie stradale fosse piana. Questa trasformazione risulta utile per risolvere ad esempio il problema introdotto in precedenza relativo all'Inverse Perspective Mapping.

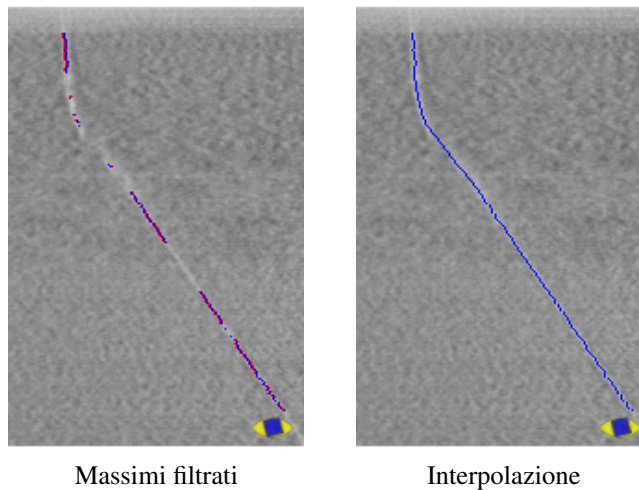


Figura 4.8: Massimi filtrati nella V-Disparity image e loro interpolazione.

Il primo passo è stato quello di interpolare i punti di massima correlazione filtrati utilizzati per tracciare il profilo del terreno. Per fare questo sono stati inseriti in un

vettore i punti in coordinate immagine ( $v,d$ ) a partire dal primo punto valido all'ultimo interpolando linearmente quelli mancanti.

Per realizzare questa operazione, si parte dall'obiettivo di ottenere dalle immagini di partenza una coppia di immagini stereo che generino una V-Disparity Image in cui la linea di correlazione sia retta. Conoscendo ora la disparità delle righe in ingresso, è possibile selezionare una loro combinazione per ottenere l'obiettivo. In particolare è definita una corrispondenza che stabilisce a quale altezza le righe dell'immagine in ingresso devono collocarsi nell'immagine d'uscita. Il risultato di questa operazione si può vedere in figura 4.9 dove alle immagini sinistra e destra è stata affiancata la corrispondente immagine priva di pendenza (in un ipotesi di superficie stradale piana).

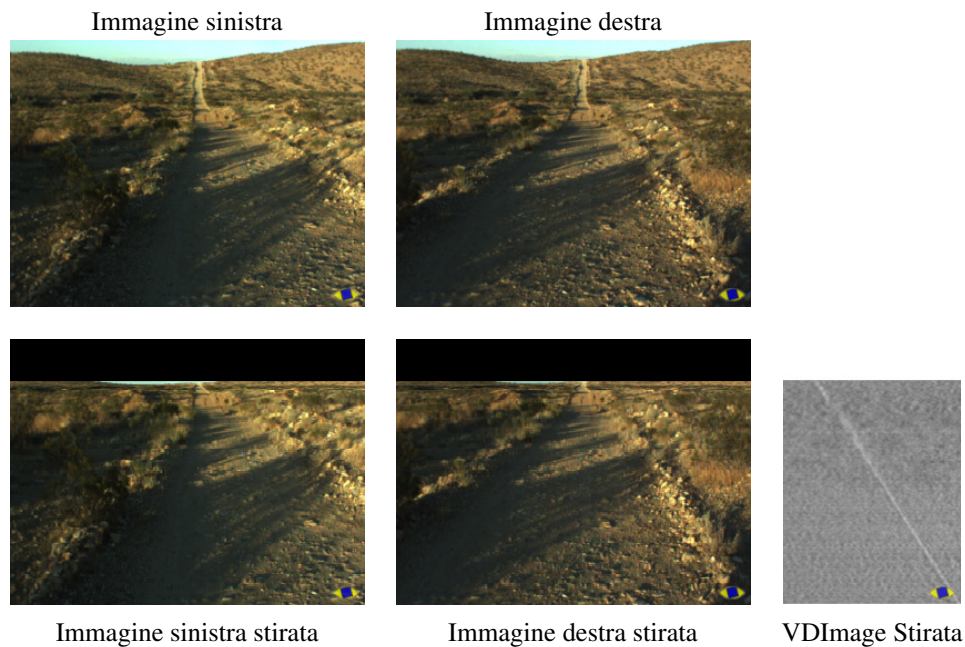


Figura 4.9: Visualizzazione delle immagini delle telecamere sinistra e destra e delle rispettive immagini stirate.

Come ultimo passo, per verifica, è possibile calcolare la V-Disparity image della coppia di immagini trovata, e come si può vedere sempre in figura 4.9, questa corrisponde ad un caso di terreno piatto. In figura 4.10 è mostrato un altro esempio di corretta rivelazione della linea di correlazione, seguita da una perfetta stiratura della superficie stradale.

### **4.3 Risultati e conclusioni**

L'algoritmo per la ricostruzione del profilo del terreno funziona correttamente quando la V-Disparity image ha buona qualità e il profilo risulta continuo grazie ai miglioramenti apportati effettuando una interpolazione lineare tra i punti. L'algoritmo per la stiratura del piano stradale fornisce inoltre risultati sorprendentemente verosimili. Sarà comunque necessario migliorare i risultati raffinando il metodo di estrazione, filtraggio e ricostruzione dei massimi per riga, e introducendo un'analisi su più frame in modo da colmare le informazioni mancanti a causa di V-Disparity image di bassa qualità.

I tempi di esecuzione dell'algoritmo implementato su un sistema a processore P4@3GHz sono risultati essere 1.2ms. Sono esclusi dal calcolo della V-Disparity image iniziale, necessaria già per altre elaborazioni, e della V-Disparity Image stirata.

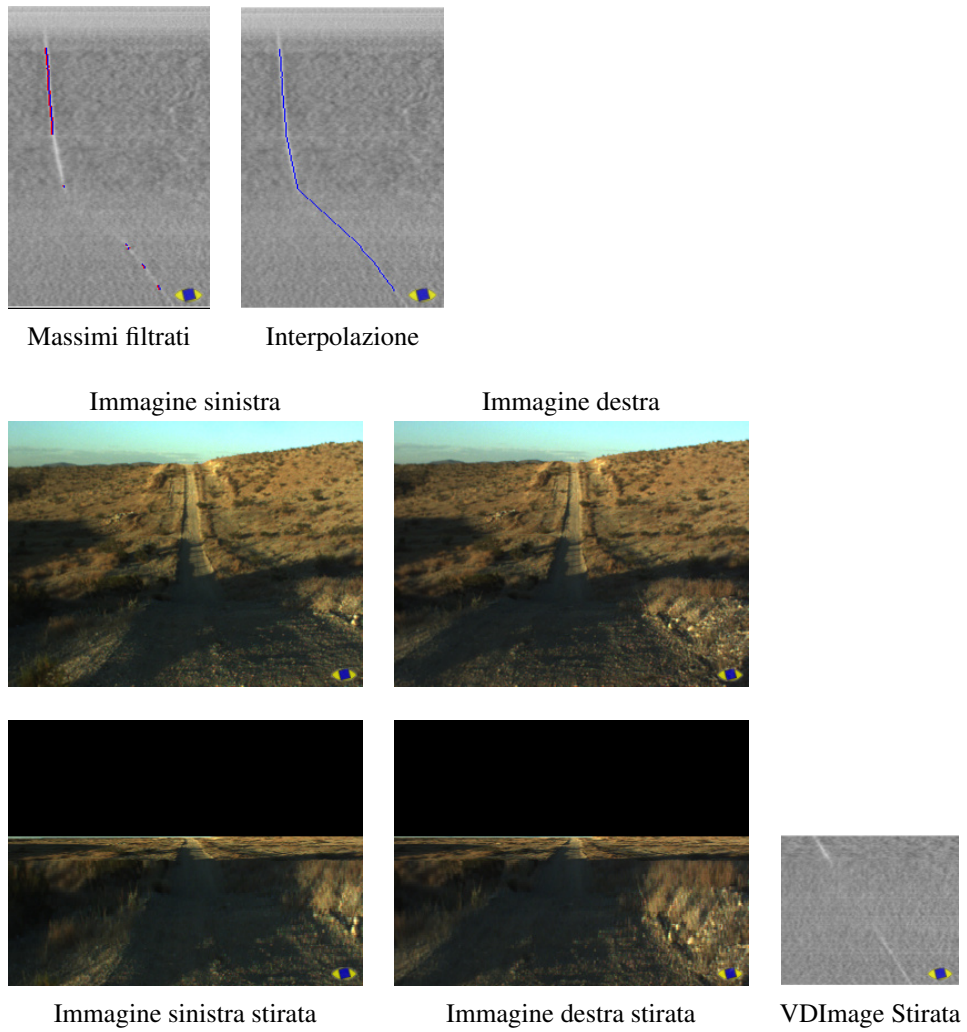


Figura 4.10: Comportamento corretto dell' algoritmo di stiratura.

## **Parte II**

# **Percezione di segnali stradali**



# Presentazione del problema

La realizzazione di un riconoscitore automatico di cartelli stradali è un importante obiettivo nell'ambito della ricerca di sistemi di supporto al guidatore allo scopo di migliorare la sicurezza sulle strade: infatti, un cartello stradale fornisce un codice visivo contenente importanti informazioni sullo stato della strada, sui pericoli o sul corretto comportamento da tenere lungo la strada.

I cartelli stradali sono solitamente colorati con colori distintivi e molti autori sono impegnati nello sviluppo di un rilevatore basato sul colore ([21, 22, 23, 24]); anche il Laboratorio di Visione Artificiale (VisLab) del Dipartimento di Ingegneria dell'Informazione dell'Università di Parma, in collaborazione con il settore Sistemi Elettronici dell'azienda Magneti Marelli, ha presentato in [18, 19] un sistema basato su immagini a colori (proveniente da una sola telecamera posizionata e orientata in modo non necessariamente preciso) e su un algoritmo a tre passi che utilizza la segmentazione dell'immagine per colore per l'individuazione delle regioni d'interesse, seguita dal riconoscimento delle forme e dall'applicazione di reti neurali per la rilevazione e classificazione di pressoché tutti i segnali stradali italiani .

L'algoritmo si avvale inoltre di un'analisi della tinta dell'asfalto nelle immagini in modo da poter risalire alla colorazione della fonte luminosa e permettere la rilevazione anche nei casi in cui condizioni particolari di luce (per es. tramonto o illuminazione artificiale con lampade al sodio) generino un cambiamento nella tinta dei cartelli.

Non tutti i cartelli hanno però colori distintivi, ed in particolare per i segnali di fine

prescrizione non è stato possibile realizzare una ricerca basata sul colore. D'altra parte la mancata rilevazione di un segnale di fine prescrizione corrisponderebbe ad avere un falso positivo: infatti un rivelatore di limite di velocità continuerebbe a considerare valido un limite terminato dal segnale di fine prescrizione, portando in breve tempo il guidatore a non porre confidenza nel sistema. Il capitolo 5, primo della seconda parte di questa tesi, riporta la trattazione della rilevazione dei cartelli di fine prescrizione, basata su un approccio totalmente diverso, costituito dalla ricerca di una forma distintiva in immagini in scala di grigio.

Nel capitolo 6, ultimo della seconda parte di questa tesi, è presa in considerazione la possibilità di aggiungere una seconda telecamera al sistema di acquisizione del rivelatore di cartelli: sfruttando le informazioni stereo derivate e mediante lo studio di altre proprietà, è affrontato il problema della rilevazione dei pannelli integrativi associati ai cartelli stradali.

## Capitolo 5

# Rilevazione di cartelli di fine prescrizione

Gli algoritmi di rilevazione di segnali stradali basati sulla ricerca del colore incontrano difficoltà nella ricerca dei segnali di fine prescrizione [22, 21]: il colore bianco di questo tipo di cartelli (si veda la figura 5.1) non costituisce una caratteristica distintiva in quanto non è differente da molti altri elementi presenti nell'immagine (ad es. l'asfalto) cosicché la ricerca per colore non è adatta all'individuazione di regioni d'interesse.



Figura 5.1: Esempi di cartelli di fine prescrizione oggetto dell'algoritmo.

Un rilevatore specifico per cartelli di forma circolare [25] potrebbe essere una soluzione a questo problema. Tuttavia l'elaborazione dell'intera immagine per la ricerca

di cerchi può essere un compito difficile e dispendioso dal punto di vista computazionale, soprattutto considerato che lo stesso sistema deve già farsi carico dell'algoritmo per la rilevazione degli altri cartelli (si vedano [18, 19]). Oltre al cerchio, un'altra caratteristica distintiva del cartello di fine prescrizione italiano è costituito da una banda nera verticale inclinata, la cui rilevazione può essere più veloce. Per questa ragione l'algoritmo presentato qui di seguito rileva le regioni di interesse identificando campioni dell'immagine compatibili con una banda nera inclinata. Quindi, dal momento che diversi elementi possono generare un campione di questo tipo, l'algoritmo ricerca attorno alle bande candidate il cerchio corrispondente ai bordi del cartello.

Per finire, l'uscita subisce un'operazione di tracking, che valida un cartello se è rilevato per almeno due frame. L'algoritmo è stato testato su immagini di risoluzione  $640 \times 480$  e  $752 \times 480$  riprese tramite una telecamera montante un sensore ad  $1/3$  di pollice e una focale di 6mm.

## 5.1 Individuazione di bande nere verticali inclinate

Per ogni riga dell'immagine in scala di grigi, la banda nera del cartello di fine prescrizione descrive una transizione chiaro-scuro-chiaro dal bianco dello sfondo del cartello al nero della banda e di nuovo al bianco dello sfondo del cartello: considerato questo la caratteristica distintiva della banda non è la sua intensità luminosa, ma la coppia di variazioni di tono di grigio ai suoi bordi. Facendo fronte ad un problema simile, [26] presenta un rivelatore di segnaletica orizzontale basato sul fatto che la riga stradale compare nell'immagine con una larghezza attesa all'interno di un piccolo range: per ogni riga dell'immagine, questo range è determinato in funzione della larghezza attesa, che dipende dalla prospettiva all'interno dell'immagine, e considerando una piccola variabilità della larghezza della riga stradale nel mondo reale. Un campione dell'immagine che rispetti queste caratteristiche è immediatamente validato. Similmente, nell'approccio proposto è considerato che la larghezza nell'immagine della banda del cartello di fine prescrizione può variare (dipende dalla distanza dalla quale è vista dalla telecamera), ma in questo caso un campione chiaro-

---

scuro-chiaro individuato può essere considerato corretto solo se la sua larghezza ha un rapporto corretto con la lunghezza della banda a cui esso sarà associato. Per questa ragione, l'algoritmo evidenzia, per ogni riga di un'immagine che contiene le variazioni di toni di grigio (immagine derivata), il centro di regioni comprese tra un picco negativo ed uno positivo, creando un'immagine HLH (High-Low-High, Alto-Basso-Alto) che contiene informazioni sulla larghezza dei campioni trovati. Le catene di campioni continui e di larghezza coerente vengono raggruppati ed esaminati in modo da decidere riguardo alla loro compatibilità con la banda di un segnale di fine prescrizione.

### 5.1.1 Ricerca di campioni High-Low-High

Per mettere in rilievo le variazioni del tono di grigio, l'algoritmo applica all'immagine un semplice filtro derivativo di forma  $(-1 \ 0 \ 1)$ , ottenendo un'immagine derivata (Es.: figura 5.2).

Per ogni riga occupata dal segnale di fine prescrizione, la banda nera genera nell'immagine derivata una caratteristica specifica formata da:

- un picco negativo, causato dal tono di grigio decrescente in corrispondenza della transizione chiaro-scuro dal bianco del cartello al nero della sua banda;
- un insieme di valori vicini a zero, generati dai pixel di valore quasi costante all'interno della banda; la larghezza di questa regione dipenda dalla distanza del cartello dalla telecamera (e dai parametri intrinseci della telecamera);
- un picco positivo, causato dal tono di grigio crescente in corrispondenza della transizione scuro-chiaro in uscita dalla banda.

Per individuare questo tipo di caratteristica l'algoritmo analizza da sinistra a destra le righe dell'immagine derivata seguendo queste regole:

- Quando individua un pixel con valore negativo, l'algoritmo crea una soglia per categorizzare i valori in negativi, vicini a zero e positivi. Questa soglia (discri-

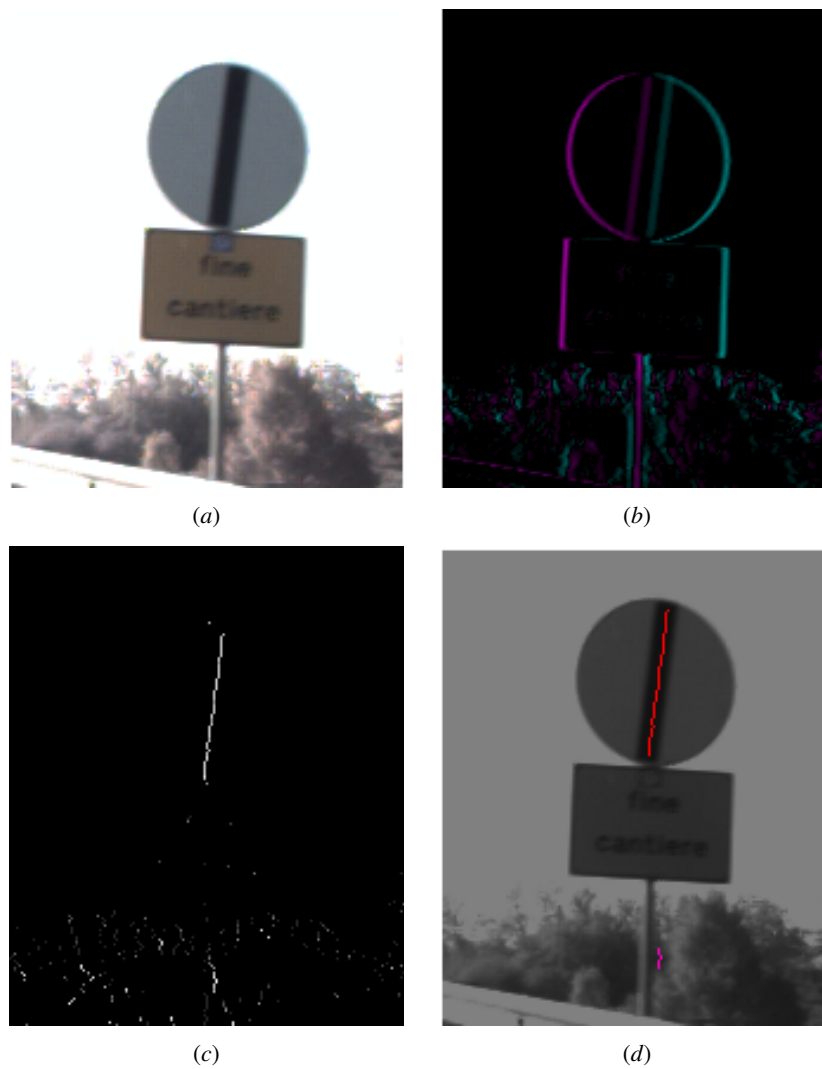


Figura 5.2: (a) Cartello di fine prescrizione da rilevare. (b) Immagine derivata: le variazioni di tono di grigio decrescenti sono evidenziate in rosso, mentre quelle crescenti in verde. (c) Immagine HLH: per una miglior visualizzazione dei risultati i valori sono normalizzati al valore più alto e moltiplicati per 255. (d) Le etichette individuate.

minative threshold) è calcolata in proporzione al valore assoluto massimo del picco ricercato in un intorno destro del primo pixel negativo.

- L'algorithmo continua quindi l'analisi della riga seguendo il diagramma a stati di figura 5.3:

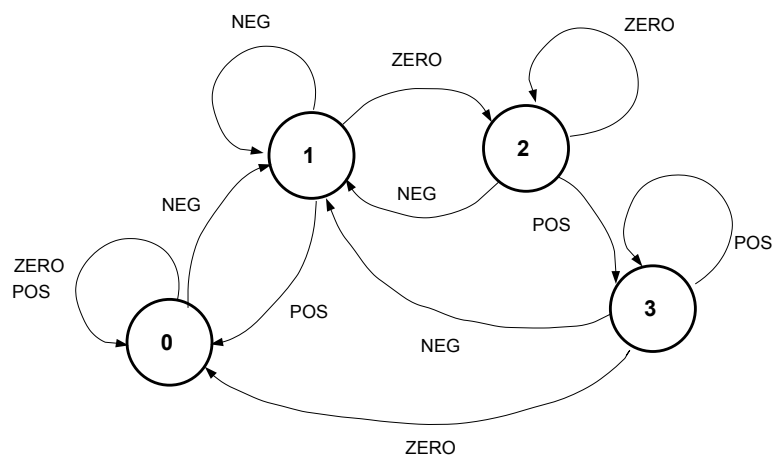


Figura 5.3: Diagramma di stato HLH.

- Dopo aver individuato il primo valore negativo e aver creato la discriminative threshold, il sistema passa dallo stato 0 allo stato 1, dove è misurata la larghezza del picco negativo.
- Quando il primo valore vicino a zero è individuato, il sistema passa dallo stato 1 allo stato 2, dove è misurata la larghezza dell'area a valori quasi costanti (la larghezza espressa in pixel della presunta banda)
- Quindi, quando è individuato un valore positivo, il sistema passa dallo stato 2 allo stato 3, dove è misurata la larghezza del picco positivo;
- Infine, quando un valore non negativo causa l'uscita del sistema dallo stato 3, il campione individuato è validato se:

- \* le transizioni di ingresso e di uscita dalla banda sono sufficientemente repentine (ovvero, le larghezze dei picchi negativi e positivi sono sotto una certa soglia);
- \* nell'immagine originale la luminosità della regione alla sinistra della banda è simile a quella della regione alla destra. Si è potuto constatare che la soglia per determinare questa similarità non può essere fissata troppo restrittiva a causa di particolari condizioni di illuminazione (si veda la figura 5.4) e dei disegni che possono essere presenti sul segnale per indicare il tipo di prescrizione terminata.



Figura 5.4: Il segnale in figura è leggermente piegato lungo il suo asse verticale; in questa particolare condizione di illuminazione il lato sinistro del cartello è in ombra, per cui la sua luminosità risulta essere diversa dal lato destro: se la soglia fissata fosse restrittiva la banda non potrebbe essere individuata.

- I campioni validati sono codificati in una nuova immagine, chiamata immagine HLH (High-Low-High, Alto-Basso-Alto), ponendo nel solo pixel in corrispondenza del centro del campione individuato il valore corrispondente alla lunghezza della sequenza dei valori vicini a zero misurati nello stato 2. In questo modo, nell'immagine HLH la banda diviene una linea verticale di valore quasi costante (figura 5.2c).

- Se la sequenza dallo stato 0 allo stato 3 non è seguita correttamente, il sistema finisce allo stato 0 o 1 senza validare il campione.

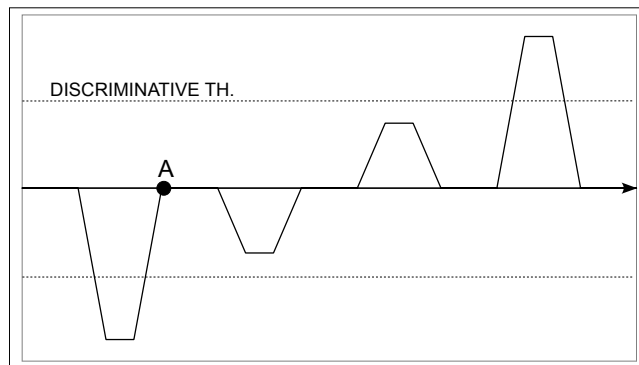


Figura 5.5: Esempio di grafico dei valori in una riga dell'immagine derivata in caso di un cartello collocato su sfondo luminoso.

- Un cartello in ombra su sfondo luminoso (figura 5.2a) genera un grafico dei valori nelle righe dell'immagine derivata con caratteristiche particolari mostrate in figura 5.5: i picchi negativi e positivi più alti sono causati dalla transizione dal cielo luminoso nello sfondo al cartello e di nuovo dal cartello al cielo. La banda del cartello di fine prescrizione causa invece due picchi minori compresi tra i due più alti. Quando il picco negativo più alto è processato, esso genera una discriminative threshold più alta dei picchi più bassi, di modo che essi sono considerati valori vicini a zero, causando la non individuazione della sequenza caratteristica della banda. Per evitare questo, quando il sistema si trova in stato 2 se il numero dei valori vicini a zero contati supera una soglia fissata il sistema ritorna in stato 0 e l'analisi ricomincia dal pixel alla destra dell'ultimo picco negativo individuato (punto A nel grafico). Lo svantaggio di questa scelta è che la banda non può essere individuata se nell'immagine compare troppo

larga, ovvero se il cartello è troppo vicino. Per la configurazione utilizzata per i test, il sistema ha dimostrato di essere in grado di rilevare la banda quando il cartello si trovava a distanze maggiori di circa 4 metri.

### 5.1.2 Etichettatura

La banda di un cartello di fine prescrizione è descritta da un'area continua scura verticale di larghezza costante. Come già anticipato, nell'immagine HLH questa caratteristica viene rappresentata da un segmento verticale di valore quasi costante. Per questa ragione, è stato implementato una funzione di etichettatura (labeling) che interrompe la crescita di una regione etichettata qualora questa si trovi a dover includere un pixel il cui valore (larghezza del campione HLH) risulti incoerente con il resto della regione etichettata. Qui di seguito è presentato come questo è stato realizzato.

La larghezza della banda di un cartello di fine prescrizione dipende dalla sua distanza dalla telecamera; dal momento che la rilevazione del segnale non può essere considerata affidabile da una distanza eccessiva, un pixel dell'immagine HLH può essere considerato "seme" di un'etichetta solo se rappresenta una larghezza della banda corrispondente ad un cartello di fine prescrizione sufficientemente vicino, ovvero se il suo valore è sopra una soglia stabilita. L'etichetta cresce iterativamente esaminando i tre pixel immediatamente sotto all'ultimo pixel aggiunto all'etichetta. Si noti che dato che la larghezza di un campione HLH è come minimo 3 per costruzione (si consideri il diagramma di figura 5.3), solo uno di questi pixel può essere acceso. Durante la fase di espansione di un'etichetta, sono raccolte le statistiche riguardanti i valori dell'immagine HLH inglobati; un nuovo pixel è aggiunto all'etichetta solo se il suo valore (larghezza del campione HLH) non differisce di più di 1 dal valore più frequente tra quelli dei pixel già inclusi nell'etichetta.

La soglia utilizzata per individuare il pixel seme non è utilizzata in fase di espansione della regione etichettata, così che una banda può essere rilevata anche se solo una sua parte rispetta la larghezza minima richiesta. Tuttavia in questo modo il pixel seme può essere individuato non alla sommità della banda, e dato che l'espansione dell'etichetta procede solo verso il basso, la parte superiore può non essere inclusa

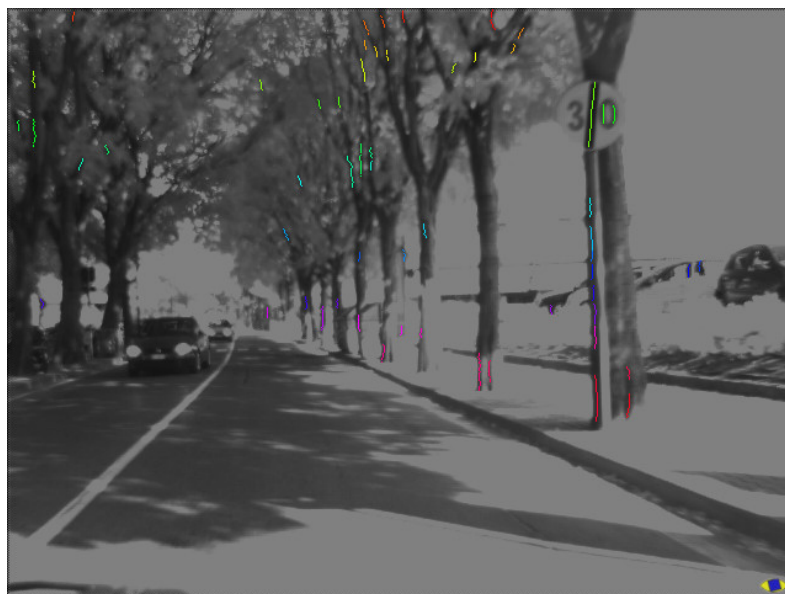


Figura 5.6: Esempio di risultato di etichettatura.

nell'etichetta. Per questa ragione, dopo aver completato l'etichettatura verso il basso, il seme dell'etichetta è fatto crescere anche verso l'alto, seguendo le stesse regole e utilizzando le statistiche già raccolte. Infine, per escludere elementi troppo corti, l'algoritmo verifica che l'area etichettata sia di dimensione superiore ad una soglia adeguata. La figura 5.2d e 5.6 mostrano due esempi di risultato di etichettatura.

### 5.1.3 Validazione e unione di regioni etichettate

Come mostra la figura 5.6, a questo punto dell'elaborazione tutti gli elementi che descrivono una banda nera (come tronchi o rami d'albero, finestre, pali, ecc.) sono etichettati come possibili bande di un segnale di fine prescrizione. Inoltre, elementi di rumore all'interno del cartello (come ad esempio l'indicazione di velocità per la quale il limite termina) possono causare un parziale mancato riconoscimento del-



Figura 5.7: Esempio di rivelazione di banda spezzata.

la caratteristica HLH precedentemente presentata; questo provoca la rivelazione di regioni aree etichettate scollegate, come mostrato nell'esempio di figura 5.7.

Per far fronte a questi problemi, l'algoritmo effettua inizialmente una verifica sul valore del rapporto  $\frac{\text{lunghezza}}{\text{larghezza}}$ : se è eccessivo l'elemento è scartato in modo da eliminare elementi troppo lunghi; se invece è contenuto in un range restrittivo (attorno ad un valore che è stato stimato essere 8), l'elemento è subito validato per la fase successiva. Per le etichette aventi rapporto  $\frac{\text{lunghezza}}{\text{larghezza}}$  giudicato piccolo sono candidate per le successive operazione di unione.

Infatti, per risolvere il problema relativo alla rivelazione di bande spezzate, le etichette vengono confrontate a coppie per verificare se è possibile unirne e ottenere bande di adeguato rapporto  $\frac{\text{lunghezza}}{\text{larghezza}}$ . Quando il punto più in alto di un'etichetta è vicino al punto più in basso di un'altra, l'algoritmo calcola la distanza, lungo l'asse  $x$ , tra il prolungamento (in linea retta) dell'etichetta posta più in basso e il punto più in basso dell'etichetta più in alto: se questa distanza è sufficientemente piccola le etichette possono essere unite. Con le etichette unite l'algoritmo verifica nuovamente

il rapporto  $\frac{\text{lunghezza}}{\text{larghezza}}$ , eventualmente validando nuovi elementi da passare ai controlli successivi.

Dato che la banda di un segnale di fine prescrizione è inclinata, l'algoritmo seleziona tra le etichette ottenute quelle che hanno adeguata orientazione, in un range attorno al valore di inclinazione ideale (che è stato stimato essere -0.12 radianti).

Giunto a questo punto, attorno ad ogni etichetta rilevata e validata l'algoritmo identifica un'area di interesse, dove verifica la presenza del cerchio causato dai bordi del cartello.

## 5.2 Individuazione di cerchi

Nel caso sotto esame è necessario stabilire un criterio che definisca la presenza di un cerchio attorno alla banda individuata, ma non è richiesto individuare nè il raggio nè la posizione del cerchio eventualmente presente. Perciò sono state applicate le seguenti semplificazioni (proprietà):

1. E' già noto dove il cerchio dovrebbe essere, perciò la regione di ricerca è limitata.
2. Si ha una stima del raggio del cerchio atteso, dato dalla metà della lunghezza della banda individuata.
3. Si presume che il centro del cerchio debba porsi al centro della banda individuata.

Sfruttando la proprietà 1, l'algoritmo individua un *bounding box* (in breve *bbox*) attorno ad ogni regione etichettata precedentemente individuata, ovvero una regione quadrata che contiene gli elementi candidati ad essere un cartello di fine prescrizione; l'algoritmo focalizza la ricerca del cerchio in quest'area; ogni *bbox* è ritagliato dall'immagine di ingresso e riscalato in modo da ottenere una piccola immagine di dimensione  $30 \times 30$  pixel. Su quest'immagine è quindi calcolata una trasformata di Hough (*immagine di Hough*) generalizzata per i cerchi (fig. 5.8). In presenza di un

cerchio l'area centrale della trasformata di Hough presenta una concentrazione di valori più alti; al contrario, quando il cerchio è assente questa concentrazione non si verifica.

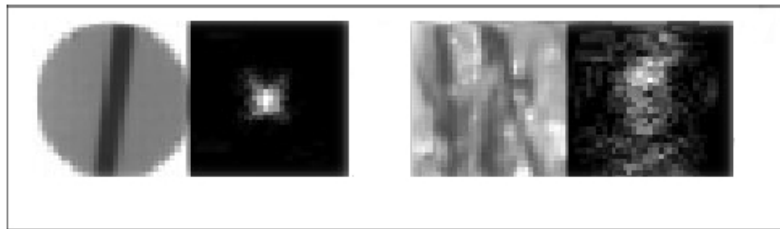


Figura 5.8: Trasformata di Hough per ricerca dei cerchi in presenza ed in assenza di un cerchio. Per una migliore visualizzazione, i valori sono disegnati normalizzando rispetto al valore più alto e moltiplicando per 255.

Diversi rilevatori di cerchi basati su differenti tipologie di trasformate di Hough sono proposti in letteratura. [27] presenta un approccio basato sulla simmetria radiale, una variante della trasformata per i cerchi di Hough che ricerca coppie di bordi orientati simmetricamente. Ogni coppia definisce completamente un possibile cerchio, e vota per la coppia cerchio-raggio coerente. Questo algoritmo è veloce (13,2 ms per immagini  $240 \times 320$  su un sistema P3@1.4GHz) e restituisce risultati di alta qualità. Come già detto, il nostro caso consente importanti semplificazioni, che si è deciso di sfruttare per aumentare la sensibilità dell'algoritmo: in particolare si è voluto estendere la rilevazione anche a casi come quello in figura 5.9 nel quale, a causa della saturazione, i bordi sono visibili solo parzialmente e un approccio basato sulla simmetria radiale incontrerebbe difficoltà.

Il rilevatore di cerchi implementato si basa su un metodo molto utilizzato in letteratura (una descrizione approfondita può essere trovata in [28]) che utilizza la fase dei bordi ottenuta tramite un filtraggio con maschere di Sobel per ridurre a due le dimensioni della trasformata di Hough. La figura 5.10 fornisce una spiegazione grafica del metodo realizzato.



Figura 5.9: Esempio di un cartello contraddistinto da bordi solo parzialmente visibili.

La scelta di una corretta soglia nel calcolo dell'immagine di Sobel riveste un'importanza critica, in quanto solo i pixel classificati come bordo contribuiscono al calcolo della trasformata di Hough. [28] presenta il problema del *tuning* (regolazione) delle soglie per l'immagine di Sobel e sulla trasformata di Hough: è molto difficile individuare un insieme di parametri che consenta la rivelazione di tutti i cerchi (anche quelli a basso contrasto rispetto allo sfondo) senza comportare la comparsa di falsi positivi. Nel caso trattato, essendo la regione di interesse molto limitata, la soglia per l'immagine di Sobel può essere fissata ad un valore particolarmente basso, consentendo alla maggioranza dei pixel di votare e mantenendo un buon rapporto  $\frac{\text{segnale}}{\text{rumore}}$ , e aumentando il numero di casi per i quali il cartello può essere rilevato (si veda la figura 5.11).

La possibilità di mantenere una bassa soglia di Sobel non sarebbe disponibile se la trasformata di Hough fosse da calcolare sull'intera immagine perché comparirebbero troppi contributi: assieme ai motivi riguardo al tempo computazionale, questa è la principale ragione per la quale la ricerca delle regioni di interesse avviene tramite l'individuazione di bande inclinate piuttosto che attraverso l'individuazione di cerchi

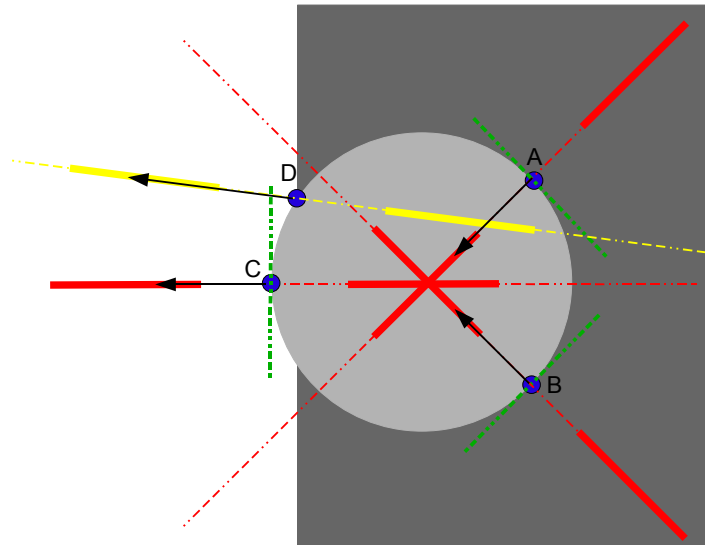


Figura 5.10: Sistema di scrutinio impiegato: ogni punto contraddistinto come bordo vota lungo una linea (tratteggio rosso) la cui direzione è data dal gradiente dell'immagine (freccia nera, calcolato tramite Sobel). Un bordo vota, come possibili centro di un cerchio che lo possa aver generato, solo per i punti nelle due sezioni della riga (segmenti rossi continui) posti ad una distanza compresa tra  $r_{min}$  e  $r_{max}$ , scelti in un intorno del raggio atteso (che è ottenuto seguendo la proprietà 2). I segmenti sono tracciati in una nuova immagine avente le stesse dimensioni utilizzando una funzione di disegno ispirata al ben noto algoritmo di Bresenham (si veda l'app. C). I segmenti che fuoriescono dai confini dell'immagine sono scartati. I punti A, B e C contribuiscono ognuno con un segmento passante per il centro, generando una concentrazione di voti. Il rumore dallo sfondo genera una rotazione del gradiente dell'immagine nel punto D, i cui voti (segmenti gialli continui) non contribuiranno alla corretta rilevazione del cerchio.

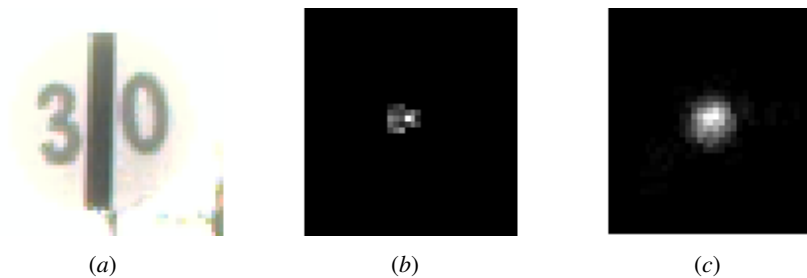


Figura 5.11: (a) Esempio di un cartello a scarso contrasto con lo sfondo. (b) Immagine di Hough ottenuta utilizzando un'alta soglia su Sobel. Non sono presenti sufficienti contributi per la determinazione del centro del cerchio: di conseguenza il cartello non potrà essere rilevato. (c) Utilizzando una soglia permissiva, l'immagine di Hough ottenuta è di buona qualità nonostante lo scarso contrasto dell'immagine di partenza.

su tutta l'immagine.

In [27] gli autori analizzano l'effetto del rumore sulla fase di Sobel (orientazione dei bordi); con il disegno di figura 5.10 vogliamo porre in rilievo un caso particolare ma significativo di questo problema: la norma di Sobel nel punto D è maggiore rispetto agli altri punti del bordo del cerchio, ma la sua fase è ruotata a causa del rumore proveniente dallo sfondo. Di conseguenza il contributo da esso apportato costituisce rumore (in fase di determinazione della presenza del cerchio) all'interno dell'immagine di Hough. Questo vuol dire che utilizzare solo i pixel a norma di Sobel più alta durante il processo di scrutinio non sempre porta a migliori risultati.

Per migliorare il rapporto  $\frac{\text{segnale}}{\text{rumore}}$ , è definita una maschera utilizzata per escludere tutti i contributi apportati dai pixel dell'immagine che si può supporre non appartengano al bordo dell'eventuale cartello: i punti appartenenti alla banda ed ad una zona centrale del cartello, dove può essere presente il disegno che indica quale prescrizione è terminata, non sono considerati. La figura 5.12 mostra un esempio di questa maschera.



Figura 5.12: Esempio di maschera applicata al *bbox* in fase di calcolo della trasformata di Hough. Essa permette di limitare i contributi di rumore apportati dalla banda e da un'eventuale disegno presente nel cartello.

Una concentrazione di voti al centro della trasformata di Hough implica la presenza del cerchio atteso. Per identificare questo, è definita un'area centrale sulla quale sono eseguiti due controlli:

- La somma dei valori nell'area centrale rapportata alla sua area in pixel deve essere superiore ad una prima soglia (è stato raccolto un numero sufficiente di voti, si veda la figura 5.13)
- Il rapporto tra la somma dei valori nell'area centrale e tra la somma dei valori nell'intera immagine di Hough deve essere superiore ad una seconda soglia (i voti raccolti sono sufficientemente concentrati, si veda la figura 5.14)

Grazie all'analisi per la ricerca dei cerchi diversi casi altrimenti impegnativi sono correttamente scartati, come ad esempio il segnale di incrocio di figura 5.15. Tuttavia le proprietà 3 e 2 non sono sempre verificate, il che causa in alcuni casi mancate rilevazioni. Nel caso del cartello di figura 5.16 la banda del segnale è parzialmente nascosta da un adesivo. La banda è comunque rilevata, ma in modo incompleto, di modo che la posizione del centro e la misura del raggio attesi non sono correttamente

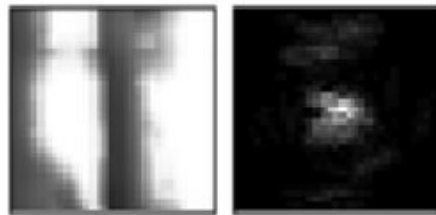


Figura 5.13: Esempio di caso particolare. Il tronco dell'albero forma una fascia nera compatibile con la banda di un cartello di fine prescrizione, mentre i rami creano un semicerchio. Tuttavia questo campione è scartato perché l'area centrale della trasformata di Hough raccoglie un numero insufficiente di voti.

stimati. Questo problema potrebbe risolto aumentando la dimensione della regione di interesse estratta intorno alla banda rilevata, ed effettuando la ricerca del cerchio con ipotesi meno restrittive sul raggio e prendendo in considerazione anche il caso di centro del cerchio posto non al centro dell'immagine di Hough.

L'intero processo finora preso in considerazione causa un numero molto piccolo di falsi positivi. D'altra parte molti fattori relativi alla qualità dell'acquisizione (come ad esempio frame in cui le immagini siano sfocate o mosse, o parzialmente saturate) possono causare una rivelazione intermittente dei cartelli di fine prescrizione.

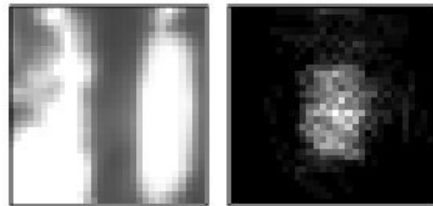


Figura 5.14: Esempio di caso particolare. Il ramo centrale genera una fascia nera compatibile con la banda di un cartello di fine prescrizione, mentre i rami attorno creano una forma semicircolare: tuttavia questo campione è scartato perché i voti al centro della trasformata di Hough non sono sufficientemente concentrati.



Figura 5.15: Esempio di corretta invalidazione tramite Hough.



Figura 5.16: Esempio di cartello di fine prescrizione la cui banda è parzialmente nascosta.

### 5.3 Tracking

Per far fronte alla rilevazione intermittente ed ai rimanenti falsi positivi, si è scelto di introdurre una semplice tecnica di tracking temporale, che non spedisce in uscita il cartello rilevato prima della sua seconda rivelazione e cancella i cartelli precedentemente individuati dopo due mancate rivelazioni consecutive. Per il frame corrente, l'algoritmo stima la posizione di un cartello precedentemente rilevato stimando la velocità nell'immagine del suo centro basandosi sulle ultime due sue rivelazioni. Un cartello è considerato nuovamente rilevato se, posto nella sua posizione stimata per il frame corrente, si sovrappone con una delle rivelazioni del frame corrente.

Si assuma che l'algoritmo abbia raggiunto una condizione di regime, per la quale ha in memoria una lista che contiene i dati relativi ai *bbox* individuati nei cicli precedenti:

- **id**: identificativo del *bbox*;
- **nDetection**: numero di frame nei quali il *bbox* è stato rilevato;
- **oldness (vecchiaia)**: da quanti cicli il *bbox* non è stato rilevato; una nuova rilevazione porta *oldness* a 0, ringiovanendo il *bbox*;
- **size**: dimensione (in pixels) del *bbox* (ovvero il lato del quadrato che lo rappresenta);
- **lastDetectedCenter**: coordinate del centro dell'ultima rivelazione del *bbox*;
- **velocity**: stima della velocità di spostamento (in  $\frac{\text{pixels}}{\text{frames}}$ ) del centro del *bbox* nell'immagine;
- **estimatedCenter**: centro stimato per il *bbox*.

All'inizio di ogni ciclo si aggiornano i dati relativi all'*oldness* e al centro stimato per ogni *bbox* (calcolato come somma tra il centro del *bbox* stimato al ciclo precedente e la velocità) contenuto nella lista che descrive la storia passata. Per poter effettuare

i confronti e calcolare correttamente le stime è necessario capire dove si trova, al frame corrente, un *bbox* rilevato in passato e confrontarlo con ogni *bbox* rilevato nel ciclo corrente; per questa ragione sono stati valutati tutti i possibili accoppiamenti tra *bbox* correnti e passati. Ad ogni coppia è assegnato un voto, stabilito sulla base del rapporto tra l'area di sovrapposizione dei due *bbox* e il massimo tra le rispettive aree, in formula:

$$voto = \frac{A_{overlap}}{\max(A_{currentBbox}, A_{pastBbox})} \quad (5.1)$$

La scelta degli abbinamenti è fatta scegliendo via via le coppie di *bbox* con voto più alto; alla fine, se un *bbox* del frame corrente non è accoppiato, significa che non era stato riconosciuto al ciclo precedente e quindi i suoi dati sono aggiunti alla lista che verrà passata al prossimo ciclo. La presenza di un abbinamento indica invece che un elemento rilevato nel frame precedente è di nuovo presente in quello corrente quindi nella lista devono essere opportunamente aggiornati i suoi dati:

- il numero di rivelazioni;
- la dimensione del *bbox*;
- il centro (sia stimato che effettivo);
- la velocità, calcolata come rapporto tra lo spostamento del centro del *bbox* (rispetto al suo ultimo centro rilevato) e la sua *oldness* ( $\frac{spostamento}{tempo}$ );
- l'*oldness*, che è portata a zero.

Gli elementi la cui *oldness* raggiunge il valore due e non trovano una nuova rivelazione sono eliminati dalla lista perché non rilevati per due frame consecutivi.

Sono invece visualizzati i *bbox* rilevati più di due volte ( $nDetection \geq 2$ ). Questo metodo permette di eliminare falsi positivi rilevati per un solo frame, per i quali *nDetection* non raggiunge il valore 2. Inoltre se l'algoritmo non riconosce nel frame corrente un segnale di fine prescrizione precedentemente individuato, grazie al tracking implementato, è possibile comunque a stimare la presenza e la posizione del

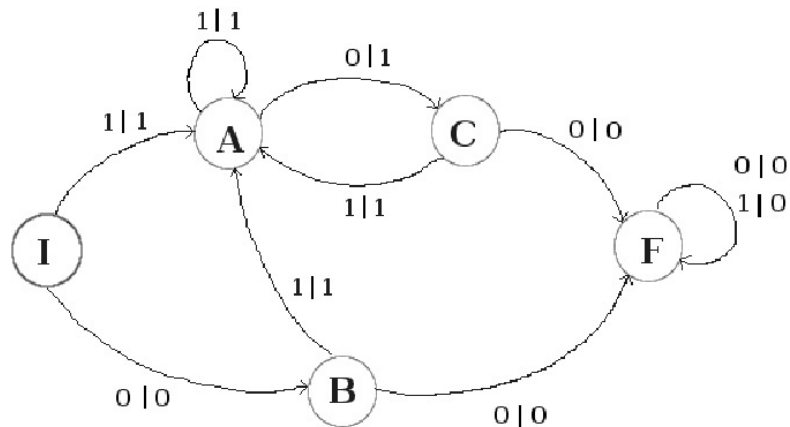


Figura 5.17: Diagramma degli stati relativo ad un *bbox*.

cartello se in due dei tre frame precedenti era stato individuato il relativo *bbox*. Inoltre in questo modo il segnale non riconosciuto può essere riassociato correttamente qualora venga rilevato correttamente al ciclo successivo.

La figura 5.17 mostra il diagramma degli stati relativo ad un *bbox* su cui agisca il tracking. In particolare:

- **I**: rappresenta lo stato iniziale del nuovo *bbox* (cioè non abbinato con gli altri elementi presenti in memoria); in questo caso *oldness* è uguale a 0 mentre *nDetection* è pari ad 1. Il *bbox* in questione non è mandato in output.
- il segnale di ingresso indica se è stato trovato o meno un abbinamento tra il *bbox* relativo al grafico ed uno di quelli rilevati al frame corrente; se l'ingresso vale 1 significa che l'abbinamento esiste e in tal caso si azzera *oldness* e si incrementa di 1 *nDetection*; se il segnale vale 0 significa che l'abbinamento non esiste quindi si incrementa *oldness* mentre *nDetection* rimane invariato;
- il segnale di uscita specifica se il *bbox* relativo al grafico è mandato in output o meno;

- **A:** è lo stato a cui passa un *bbox* dopo aver trovato un abbinamento con un elemento rilevato nel frame corrente. In questo caso *oldness* vale 0 mentre *nDetection* è almeno 2: pertanto il *bbox* è mandato in output fino alla sua cancellazione;
- **B:** è lo stato a cui passa un *bbox* che nel ciclo subito dopo la prima rivelazione non è stato accoppiato ad alcun *bbox* nuovo; *nDetection* è ancora 1, e il *bbox* non è visualizzato; *oldness* è 1 una nuova mancata rivelazione porterà alla cancellazione del *bbox*, senza che questi sia mai mandato in uscita;
- **C:** è lo stato a cui passa un *bbox* che si trova già in fase di visualizzazione ( $nDetection \geq 2$ ) ma nel frame corrente non è stato accoppiato ad alcun *bbox* nuovo; in questo caso *oldness* vale 1. Da notare come il *bbox* sia comunque mandato in output;
- **F:** è lo stato in cui un *bbox* finisce se non è rilevato per 2 frame consecutivi: esso non è più mandato in output, indipendentemente dal fatto che successivamente si trovi o meno un abbinamento con nuovi *bbox*; a questo punto il *bbox* in questione può essere cancellato.

La figura 5.18 mostra un esempio del funzionamento del tracking, nel quale l'alta velocità di guida causa un movimento rapido del cartello nell'immagine. In figura 5.18a il cartello è riconosciuto per la prima volta. Il cartello è mandato in output quando rilevato per la seconda volta (fig. 5.18b). La sovrapposizione in questo caso è possibile perché il segnale è ancora lontano e il suo movimento nell'immagine è sufficientemente piccolo. Tuttavia, a questo punto, è possibile stimare la velocità (dal movimento misurato nell'immagine) e la posizione attesa nel prossimo frame: senza questo accorgimento non sarebbe possibile ottenere l'abbinamento dei *bbox* nei passaggi da fig. 5.18c a fig. 5.18d, e da fig. 5.18d a fig. 5.18e, per i quali i movimenti del cartello sono maggiori alla dimensione del *bbox* associato. Nella figura 5.18f, il cartello è troppo vicino alla telecamera (si veda il paragrafo 5.1) e non è quindi più rilevabile. Tuttavia la posizione del segnale è ricostruita con buona approssimazione;

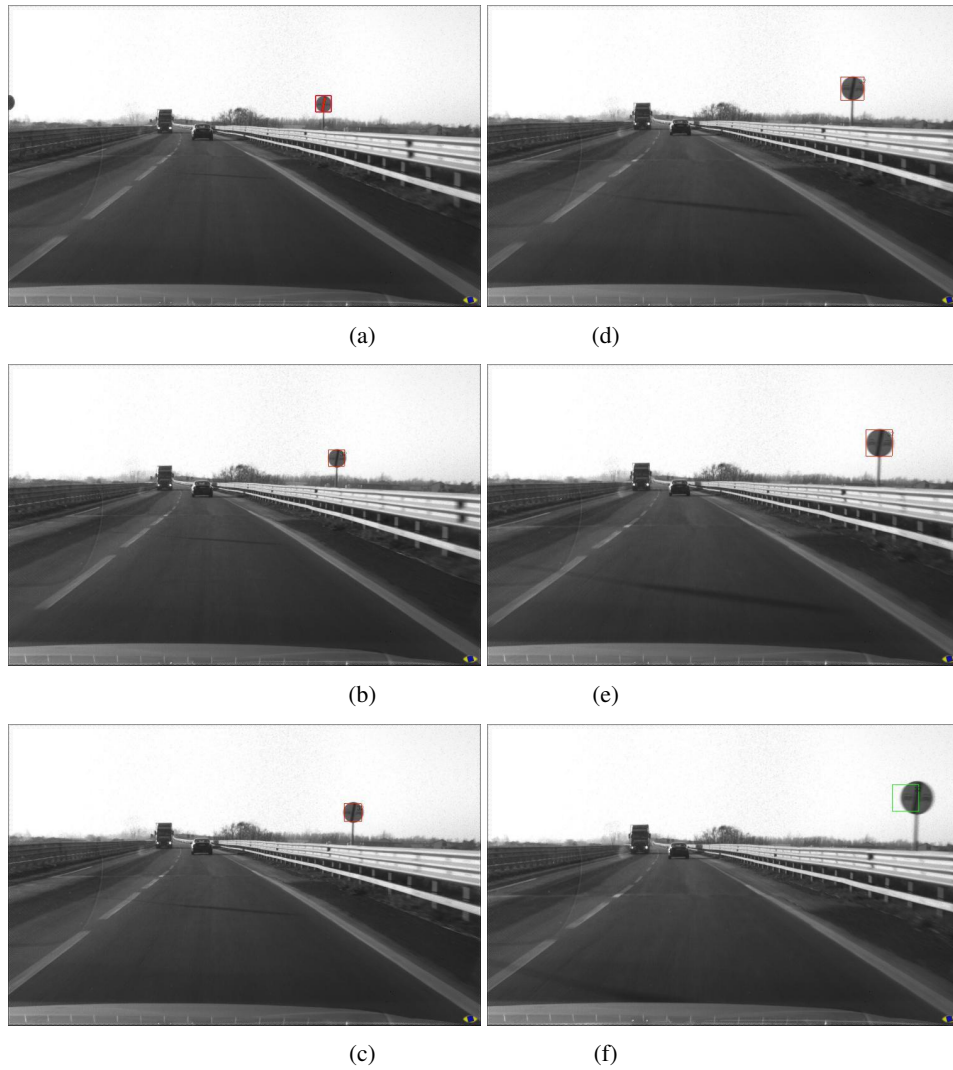


Figura 5.18: Esempio di tracking. Le immagini sono prese ad un frame rate di 12 Hz.

il *bbox* è conservato per un frame e mandato in output. La mancata rivelazione al prossimo frame causerà la definitiva cancellazione del *bbox*.

## 5.4 Risultati

Testato su più di 10000 frames ripresi in condizioni di luce diurna, l'algoritmo ha dimostrato le sue potenzialità rilevando 35 cartelli di fine prescrizione su 37 e generando 2 falsi positivi. Una rivelazione significa che il cartello è trovato per almeno 2 frame consecutivi (o al massimo separati da una frame di mancata rivelazione). I 2 cartelli non rilevati sono quello con la banda parzialmente nascosta (fig. 5.16) e quello mostrato in figura 5.19c, che, non avendo nessun contrasto con lo sfondo, non presenta un cerchio visibile nelle immagini.

La rilevazione ha successo con le diverse tipologie di cartello di fine limite testate (figure 5.19a, 5.19b, 5.19d, 5.19g e 5.19o). Le figure 5.19p e 5.19q sono esempi dell'affidabilità del sistema in presenza di elementi (in questo caso alberi) che generano campioni compatibili con una banda di un segnale di fine prescrizione.

Il rilevatore ha successo se c'è contrasto tra il cartello e lo sfondo (anche nel caso di contrasto lieve come in figura 5.19b); se i bordi del cartello non sono visibili, l'individuazione fallisce (figura 5.19c). La rivelazione è tollerante a immagini di input rotate (ruotate), anche se questo causa una banda inclinata in modo diverso da quanto atteso (figura 5.19d); questa figura mostra inoltre un ulteriore caso di rivelazione corretta in presenza di un cartello posto parzialmente su sfondo saturato. L'algoritmo è inoltre in grado di rilevare un cartello di fine prescrizione sotto diverse condizioni di illuminazione e ombra, come nei casi mostrati nelle figure 5.19f, 5.19m e 5.19n.

I tempi computazionali sono stati misurati su un sistema a processore P4@2.8GHz, processando immagini a risoluzione 640×480 images, e suddividendo e cronometrando separatamente i principali task dell'algoritmo:

- *HLH*: calcolo dell'immagine derivata e dell'immagine HLH;
- *Etichettatura*: funzione di etichettatura e validazione dei segmenti individuati;
- *Hough*: ricerca del cerchio.

Per i primi due task il tempo è costante: 6 ms per HLH e 2 ms per l'etichettatura. Il costo computazionale del terzo task dipende dal numero di etichette che devono



Figura 5.19: Rilevazione di cartelli di fine prescrizione: con contrasto del cartello sullo sfondo alto (a) e basso (b); (c) in assenza di contrasto con lo sfondo il cartello non può essere rilevato; (d) in un immagine rollata, in presenza di sfondo parzialmente saturato; (e) in un immagine scura; (f) ombre parziali; (g-h) alta velocità di guida; (i) immagine mossa; (continua)

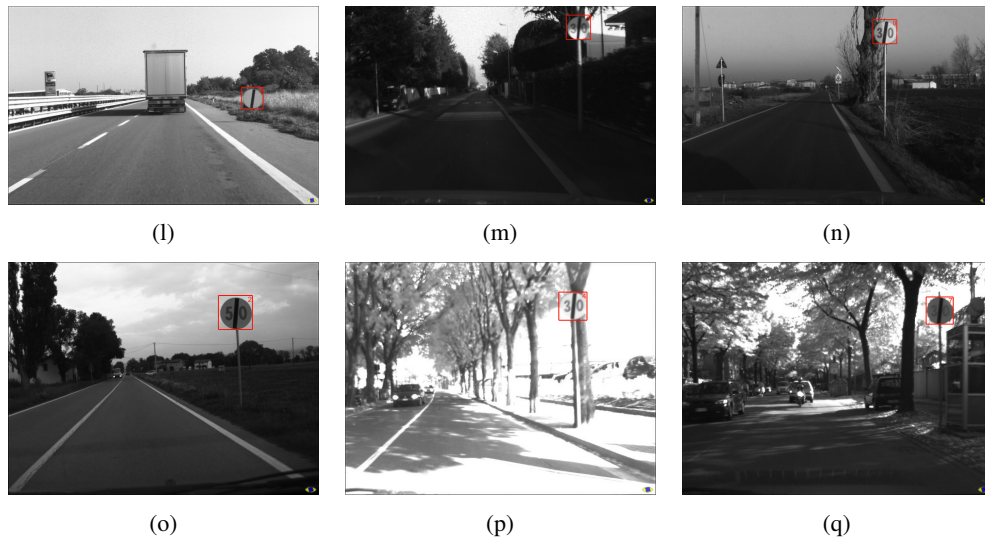


Figura 5.19: (*continua*) (l) cartello appoggiato per terra; (m) parte superiore e inferiore del cartello illuminate in modo differente; (n) lato destro e sinistro illuminati in modo diverso; (o) fine limite 50km/h; (p) sfondo parzialmente saturato; (p)-(q) in presenza di alberi.

essere processate con la trasformata di Hough: per via del ricampionamento a immagini  $30 \times 30$  pixel, il tempo è approssimativamente costante per ogni rivelazione di cerchio ( $450 \mu s$ ); in un frame computazionalmente dispendioso come quello mostrato in figura 5.6, dove il numero di etichette da processare è 9, il tempo richiesto è stato di 4 ms.

## 5.5 Considerazioni

Mentre i segnali di prescrizione e di pericolo sono stati progettati per essere facilmente visibili, i cartelli di fine prescrizione non necessitano immediata attenzione dal guidatore, cosicché non è stata attribuita nessuna caratteristica di colore peculiare. La loro rivelazione diventa quindi particolarmente difficile, perché è difficile realizzare l'individuazione di regioni di interesse nell'immagine. E' stato perciò deci-

so di affrontare questo problema in modo focalizzato, realizzando però un algoritmo estraneamente specializzato su questo singolo compito. Altri approcci più generali per l'individuazione di regioni di interesse sono proposti in letteratura, come rilevatori di bianco ([21, 29]) e basati sulla ricerca di cerchi. Tuttavia, nessuno di essi sembra essere sufficientemente indipendente dalle condizioni di illuminazione.

L'algoritmo è stato testato su immagini impegnative, in modo da controllare le sue potenzialità sotto diverse condizioni di illuminazione ed anche su immagini difettose. Per ottenere una rivelazione quasi completa, le soglie sono state mantenute molto basse, causando la comparsa di qualche falso positivo. Tuttavia, se è possibile scartare il caso di immagini scure, mosse o saturate (condizioni che dipendono dalla bontà del sistema di acquisizione), le soglie possono essere mantenute più alte rimuovendo questi falsi positivi. Ciononostante, i test devono essere estesi su un numero maggiore di frame, includendo anche immagini notturne.

In Europa, la maggior parte delle prescrizioni iniziate da un cartello non sono terminate da un pannello di fine prescrizione: infatti, la presenza di un incrocio con un'altra strada termina la prescrizione se il segnale di prescrizione non è ripetuto dopo l'incrocio. E' interessante osservare come per questa ragione, cartelli indicanti incroci e rotoarie definiscono il punto in cui la prescrizione in atto termina. Tuttavia questo problema non sembra affrontabile interamente con il solo uso della visione artificiale: anche se si fosse in possesso di un perfetto rivelatore di incroci, sarebbe praticamente impossibile distinguere un incrocio normale da un ingresso ad una strada privata (che invece non fa terminare la prescrizione). L'utilizzo di mappe e GPS possono aiutare nel localizzare degli incroci e nel determinare il valore del limite di velocità in assenza di cartelli di limite basato sulla posizione nel mondo del veicolo (ad esempio se si trova in città, su strada extraurbana o in autostrada).

## 5.6 Conclusioni

In questa trattazione abbiamo descritto un algoritmo progettato specificatamente per riconoscere cartelli di fine prescrizione italiani, che non sono facilmente rilevabili

tramite un approccio basato sul colore. L'algoritmo ha avuto buoni risultati nei test realizzati, con un'elevata percentuale di rivelazioni e un numero limitato di falsi positivi generati. Il calcolo è sufficientemente veloce da poter essere portato a termine sullo stesso sistema dove un altro rilevatore di segnali basato sul colore esegue i suoi calcoli. Il range di distanze per il quale il cartello può essere rilevato dipende strettamente dal sistema di acquisizione utilizzato (per esempio dalla lunghezza focale); tuttavia un cartello di fine prescrizione non necessita di essere comunicato al guidatore con particolare prontezza.

Lo studio di questo caso specifico ha mostrato nuove opportunità che in parte guideranno i futuri sviluppi anche del rilevatore basato sul colore, per esempio introducendo l'analisi dell'immagine delle fasi di Sobel per una migliore rivelazione delle forme e un miglior ritaglio delle regioni di interesse. Gli sviluppi futuri del rilevatore di cartelli di fine prescrizione riguarderanno la rimozione delle forti assunzioni riguardo alla lunghezza del raggio e alla posizione del centro durante la ricerca della forma circolare, l'applicazione di reti neurali per rimuovere i falsi positivi rimanenti e per riconoscere la specifica limitazione terminata dal segnale di fine prescrizione, e l'estensione dell'algoritmo per rilevare anche segnali di fine prescrizione di altri stati (si veda la figura 5.20).



Figura 5.20: Esempi di cartelli di fine prescrizioni spagnoli (sinistra) e tedesco (destra).



## Capitolo 6

# Riconoscimento di pannelli integrativi

Se invece di una singola immagine si utilizza una coppia di immagini stereo per la percezione dei cartelli stradali, è possibile sfruttare questo maggiore contenuto informativo per considerazioni geometriche sul mondo reale. In particolare, per un cartello è possibile conoscere la distanza a cui esso si trova rispetto alle telecamere, e quindi utilizzare le sue dimensioni reali come criterio per una sua validazione. In questo capitolo sarà esaminato un altro impiego della visione stereo applicata alla ricerca dei segnali stradali, consistente nella rilevazione dei pannelli integrativi associati ai cartelli stessi.

Il sistema presentato in [18, 19] ricerca le regioni delle immagini in cui sono contenuti i cartelli stradali. Un cartello individuato è rappresentato in uscita come un rettangolo detto *bounding box* (o in breve *bbox*), che individua la posizione e la dimensione del cartello nell'immagine di destra della coppia stereo. Unitamente al *bbox* sono anche disponibili informazioni riguardo alla categoria (forma e colore principale) di cartello individuato. Il compito dell'algoritmo trattato in questo capitolo è valutare se le regioni dell'immagine attorno ai *bbox* (in uscita dal sistema di ricerca dei cartelli) contengono pannelli integrativi.

Prima di cominciare la trattazione è doveroso precisare che tutte queste possibilità vengono però con tre principali svantaggi:

- un maggiore costo del sistema realizzato;
- una difficoltà maggiore per il montaggio del sistema;
- la necessità di affrontare il problema della calibrazione, che invece per il sistema di percezione a singola camera non ha mai costituito un grosso ostacolo.

### 6.1 Caratteristiche dei pannelli integrativi e scelta delle regioni di ricerca

Come mostra la figura 6.1, i pannelli integrativi possono essere divisi in 4 categorie:

- pannelli integrativi “piccoli”<sup>1</sup> posti al di sopra del cartello;
- pannelli integrativi “piccoli” posti al di sotto del cartello;
- pannelli integrativi “grandi” posti al di sotto del cartello in questione;
- pannelli integrativi “grandi” posti sulla destra del cartello in questione.



Figura 6.1: Esempi delle 4 tipologie di pannelli integrativi.

E' inoltre possibile fare le seguenti considerazioni riguardo alla posizione attesa per il pannello integrativo:

<sup>1</sup>Un pannello integrativo è considerato “piccolo” o “grande” in proporzione alle dimensioni del cartello a cui è associato

- i cartelli triangolari (“pericolo”) possono essere solo pannelli integrativi sottostanti e, dualmente, un segnale a forma di triangolo ribaltato (“dare la precedenza”) può avere solo espansioni soprastanti;
- l’unica tipologia di segnali che può avere pannelli integrativi sulla destra sono quelli a forma circolare (ad esempio un cartello di divieto d’accesso con indicazione dei tipi di veicoli a cui è permesso il transito).

Essendo noto in ingresso il tipo di forma del cartello, è quindi possibile evitare la ricerca di pannelli integrativi che non rispettino le regole appena presentate.

Per ogni cartello in ingresso l’algoritmo rileva quindi delle regioni di interesse (*espansioni*) che si pongono al di sotto, al di sopra ed alla destra del cartello, in particolare:

- Al di sopra del cartello, una regione di altezza pari ad  $1/3$  del bbox (e stessa larghezza del bbox).
- Al di sotto del cartello, tre regioni (denominate *prima*, *seconda* e *terza*) di altezza pari ad  $1/3$  del bbox (e stessa larghezza del bbox). La validazione di un’eventuale pannello integrativo “grande” sottostante al cartello avviene se più di una di queste regioni di interesse sono validate contemporaneamente. Se solo la regione più vicina al cartello è validata, il pannello integrativo individuato è considerato “piccolo”.
- Alla destra del cartello, una regione di larghezza pari a  $3/2$  del bbox (e stessa altezza del bbox).

## 6.2 Utilizzo della visione stereo

La proprietà che si vuole utilizzare in primo luogo è che un pannello integrativo si trova rispetto alle telecamere alla stessa distanza del cartello a cui appartiene. Utilizzando le librerie stereo disponibili, l’algoritmo calcola quindi la disparità del cartello (impiegando finestre di confronto della stessa dimensione del cartello), e verifica che le espansioni presentino una disparità compatibile. La disparità restituita è un valore

in virgola mobile (*subpixel disparity*) e il range di compatibilità è stato fissato come il massimo tra il 5% della disparità del cartello ed 1.

Per realizzare il calcolo, sono ritagliate dall'immagine destra la regione di interesse e dall'immagine sinistra l'area corrispondente per la quale saranno realizzati i confronti necessari per ottenere la disparità: il ritaglio dall'immagine sinistra sarà quindi largo quanto la regione di interesse più la massima disparità ammessa. La disparità è calcolata utilizzando le immagini dei bordi calcolata tramite un filtro di Sobel verticale applicato sull'immagine in scala di grigi. Le finestre di confronto impiegate ha come dimensioni le stesse della regione di interesse.

Nel caso delle regioni a destra dei cartelli occorre prendere in considerazione un fattore messo in evidenza dalle immagini di figura 6.2: in questo caso, la regione di



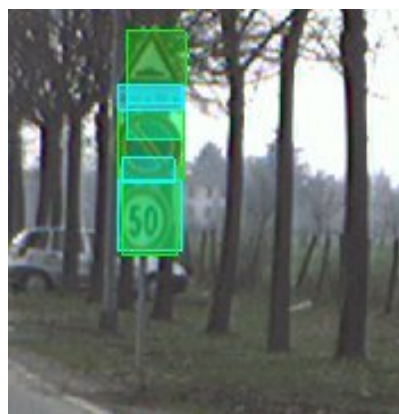
Figura 6.2: Calcolo della disparità errato dovuto all'occlusione (da notare anche 2 falsi positivi causati dal palo di sostegno, la cui trattazione avverrà nel paragrafo 6.3).

interesse a destra del cartello non contiene un pannello integrativo, ma elementi distanti che nel ritaglio dall'immagine di sinistra sono parzialmente occlusi dal cartello stesso. Il cartello causa inoltre un significativo contributo nell'immagine dei bordi. Senza nessun accorgimento il calcolo della disparità sarebbe pesantemente inficiato da questo contributo di rumore, che paradossalmente porta ad ottenere in molti casi un valore di disparità pari proprio a quello del cartello, in quanto tale disparità è la prima che permette nel confronto di non prendere in considerazione questa interfe-

renza (la finestra di confronto nell'immagine di sinistra non si sovrappone al bordo del cartello).

Per ovviare al problema si è scelto di distanziare dal cartello la regione di interesse destra di uno spazio pari alla disparità calcolata per il cartello; in questo modo l'area occlusa non è considerata nel calcolo. Questa soluzione rappresenta un'alternativa rispetto alla scelta, che, anche se più precisa, avrebbe richiesto la modifica delle librerie stereo utilizzate, di prendere l'immagine sinistra come riferimento per cercare l'omologo nell'immagine destra.

Nel caso di cartelli incolonnati sullo palo di stesso sostegno (fig. 6.3), per ogni cartello le espansioni possono sovrapporsi con le aree occupate dagli altri cartelli. In questo caso le espansioni sono erroneamente validate perché la disparità individuata corrisponde con quella del cartello che le ha generate (i cartelli incolonnati presentano la stessa disparità). Per ovviare a questo possibile problema, l'algoritmo scarta le espansioni che si sovrappongono con uno degli altri cartelli ricevuti in ingresso.



Prima del controllo di sovrapposizione



Dopo il controllo di sovrapposizione

Figura 6.3: Controllo di sovrapposizione tra espansioni e altri cartelli. Le espansioni validate sono evidenziate in azzurro.

Nonostante questi accorgimenti, il criterio della disparità non può essere considerato come l'unico per decidere sulla validità di un'espansione, in quanto altri elementi

diversi da un pannello integrativo possono causare una disparità analoga a quella del cartello. Un esempio di questo effetto è visibile in figura 6.4.

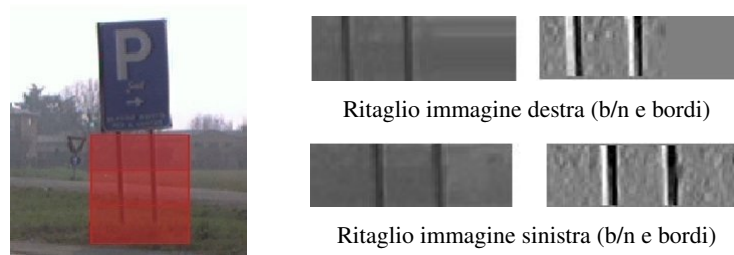


Figura 6.4: I sostegni del cartello generano nella regione di interesse bordi verticali molto significativi che incidono sul calcolo della disparità: la disparità trovata è la stessa del cartello. I ritagli si riferiscono alla prima espansione.

## 6.3 Elaborazione delle funzioni relative ai bordi

### 6.3.1 Elaborazione dei bordi verticali

Come è stato appena presentato, pali e supporti del cartello, poiché incidono significativamente nell'immagine dei bordi e sono posti alla medesima distanza del cartello dalla telecamera, possono essere causa di un valore di disparità compatibile a quello calcolato per il cartello. Partendo dalla considerazione che, di norma, un'espansione valida ha bordi verticali diffusi, determinati ad esempio da scritte contenute al suo interno (figura 6.6), mentre un sostegno ha bordi verticali più concentrati (figura 6.5), è stato aggiunto un criterio per la validazione dei pannelli integrativi sottostanti e soprastanti che considera le funzioni che descrivono i bordi verticali delle immagini. Al fine di concentrare l'analisi sul piano del cartello (*foreground*) ed evitare che elementi dello sfondo (*background*) portino un contributo errato, è stato scelto di valutare, l'istogramma per colonne dei bordi verticali, sia per i ritagli dall'immagine destra sia per quelli dall'immagine sinistra delle espansioni validate tramite il criterio della

disparità, ed individuare le zone del grafico ricavato che dessero un contributo coerente secondo la disparità calcolata. Per motivi pratici, si è scelta un'elaborazione che al termine del confronto porta ad un'unica funzione su cui eseguire le elaborazioni successive.

In particolare i ritagli sono inizialmente filtrati tramite *Sobel* verticale, mantenendo l'informazione sul segno; sono poi sommati per colonne gli elementi dell'immagine ricavata, ottenendo così, per ogni ritaglio, la funzione che descrive l'andamento per colonna dei bordi verticali (*Istogramma dei bordi verticali*). In presenza di pali e supporti nel ritaglio destro e sinistro di un presunto pannello integrativo, le funzioni associate ai bordi verticali hanno, di norma, un andamento come quello mostrato in figura 6.5: la serie di valori vicini allo zero evidenzia la mancanza di bordi verticali significativi (sullo sfondo e in corrispondenza del supporto); il picco positivo denota la presenza di un bordo verticale forte in corrispondenza del bordo sinistro di ogni palo, mentre il picco negativo e la variazione di segno sono causati dal bordo destro del palo. La figura 6.6 mostra invece l'andamento dell'istogramma dei bordi verticali relativamente ad un pannello integrativo valido: i picchi diffusi sono generati dai bordi verticali generati dalle scritte.

Per confrontare correttamente le due funzioni (istogramma dei bordi verticali dei ritagli destro e sinistro) è stato necessario traslare la funzione sinistra di una quantità pari alla disparità; mentre gli istogrammi sono definiti per valori interi, la disparità è un valore a virgola mobile quindi la traslazione ha richiesto un ricampionamento; la funzione sinistra traslata e ricampionata è determinata approssimando all'intero più vicino ogni ascissa traslata e calcolando il valore della funzione in un intorno di ogni ascissa traslata, attraverso l'applicazione di tecniche di interpolazione basate sulla valutazione della parabola passante per tre punti. In particolare: dati tre punti  $(X_1, Y_1)$ ,  $(X_2, Y_2)$  e  $(X_3, Y_3)$  i coefficienti della parabola di equazione:

$$y = ax^2 + bx + c \quad (6.1)$$

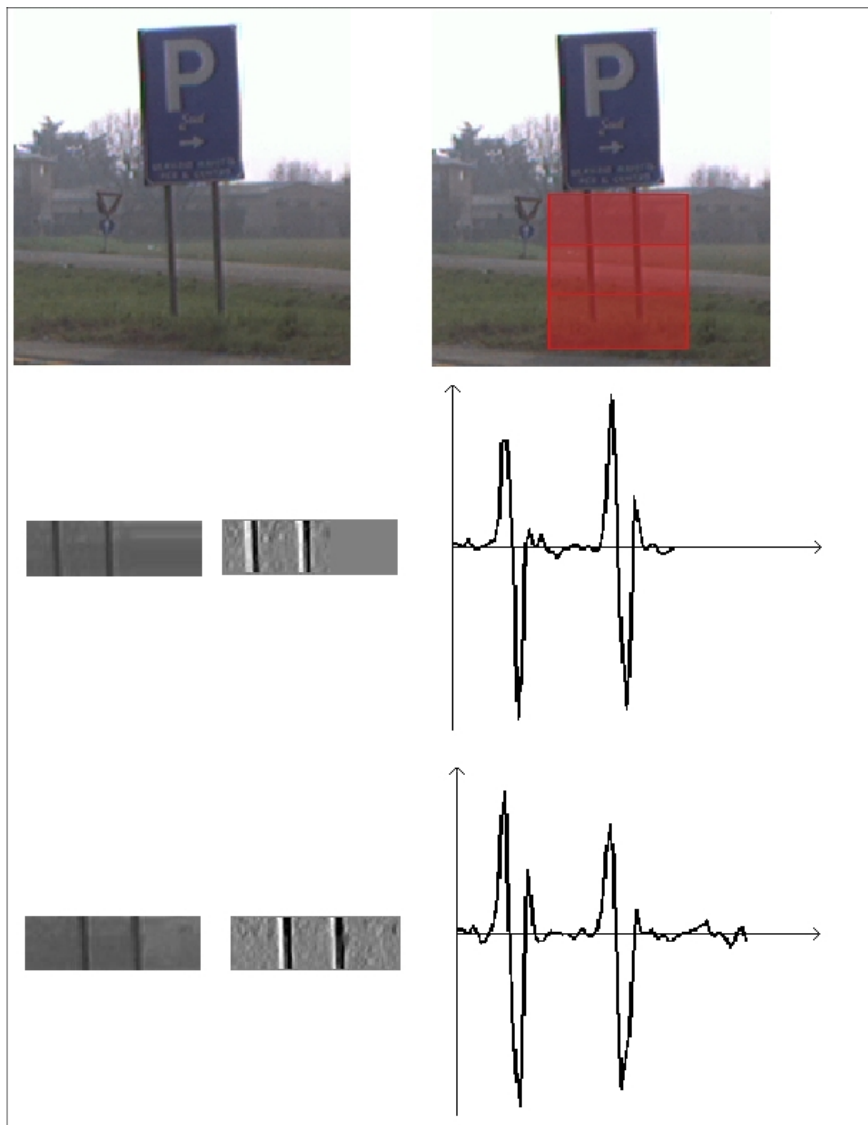


Figura 6.5: Istogrammi dei bordi verticali relativi ai ritagli destro e sinistro (traslato della disparità) di un presunto pannello contenente un doppio elemento di supporto.

I risultati si riferiscono alla prima espansione sottostante al cartello.

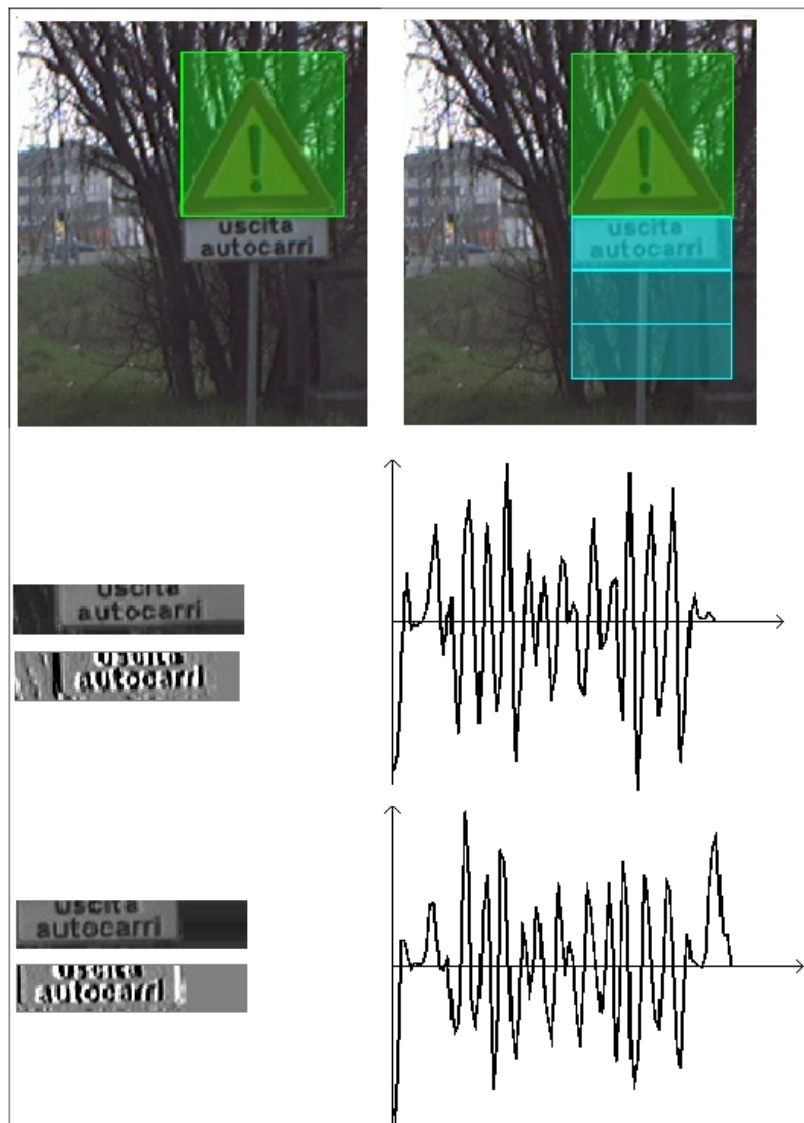


Figura 6.6: Istogrammi dei bordi verticali relativi ad un pannello integrativo valido. Nel caso in esame sono validati anche i due terzi sottostanti al pannello valido perché lo sfondo introduce rumore e porta alla generazione di picchi significativi nella funzione che descrive i bordi.

passante per essi sono:

$$a = \frac{X_1 \times (Y_2 - Y_3) + X_2 \times (Y_3 - Y_1) + X_3 \times (Y_1 - Y_2)}{(X_2 - X_1) \times (X_3 - X_1) \times (X_3 - X_2)} \quad (6.2)$$

$$b = -\frac{X_1^2 \times (Y_2 - Y_3) + X_2^2 \times (Y_3 - Y_1) + X_3^2 \times (Y_1 - Y_2)}{(X_2 - X_1) \times (X_3 - X_1) \times (X_3 - X_2)} \quad (6.3)$$

$$c = -\frac{X_1^2 \times (X_2 \times Y_3 - X_3 \times Y_2) + X_1 \times (X_3^2 \times Y_2 - X_2^2 \times Y_3) + X_2 \times X_3 \times Y_1 \times (X_2 - X_3)}{(X_2 - X_1) \times (X_3 - X_1) \times (X_3 - X_2)} \quad (6.4)$$

Se i 3 punti hanno coordinata x crescente con intervallo 1 (es.:  $X_1 = 5.4$ ,  $X_2 = 6.4$ ,  $X_3 = 7.4$ ), il denominatore vale 2.

$$a = \frac{X_1 \times (Y_2 - Y_3) + X_2 \times (Y_3 - Y_1) + X_3 \times (Y_1 - Y_2)}{2} \quad (6.5)$$

$$b = -\frac{X_1^2 \times (Y_2 - Y_3) + X_2^2 \times (Y_3 - Y_1) + X_3^2 \times (Y_1 - Y_2)}{2} \quad (6.6)$$

$$c = -\frac{X_1^2 \times (X_2 \times Y_3 - X_3 \times Y_2) + X_1 \times (X_3^2 \times Y_2 - X_2^2 \times Y_3) + X_2 \times X_3 \times Y_1 \times (X_2 - X_3)}{2} \quad (6.7)$$

Considerando il caso in cui  $X_1 = X_2 - 1$  e  $X_3 = X_2 + 1$ <sup>2</sup> e semplificando ulteriormente centrando i punti in zero, ponendo quindi  $X_1 = -1$ ,  $X_2 = 0$  e  $X_3 = 1$ , rimane:

$$a' = \frac{(Y_3 + Y_1)}{2} - Y_2 \quad (6.8)$$

$$b' = \frac{(Y_3 - Y_1)}{2} \quad (6.9)$$

$$c' = Y_2 \quad (6.10)$$

che saranno quindi i coefficienti della funzione della parabola passante per i tre punti:

$$y = a'(x - X_2)^2 + b'(x - X_2) + c' \quad (6.11)$$

<sup>2</sup>Sono perciò esprimibili come  $(X_2 - 1, Y_1)$ ,  $(X_2, Y_2)$  e  $(X_2 + 1, Y_3)$

Supponendo ad esempio che:

- la funzione sinistra passi per punti  $P_1(-1, 0)$ ,  $P_2(0, 5)$ ,  $P_3(1, 4)$ ;
- la disparità sia uguale a 3.8;

i coefficienti della parabola passante per i punti in questione risultano essere:

$$a' = \frac{(Y_3 + Y_1)}{2} - Y_2 = \frac{(4 + 0)}{2} - 5 = -3 \quad (6.12)$$

$$b' = \frac{(Y_3 - Y_1)}{2} = \frac{(4 - 0)}{2} = 2 \quad (6.13)$$

$$c' = Y_2 = 5 \quad (6.14)$$

quindi l'equazione della parabola diventa:

$$y = a'(x - X_2)^2 + b'(x - X_2) + c' = -3(x - 0)^2 + 2(x - 0) + 5 = -3x^2 + 2x + 5 \quad (6.15)$$

Con la disparità pari a 3.8, il punto  $P_2(0, 5)$  genera un punto con ascissa uguale a 4, per il quale però bisogna prendere il valore della funzione in  $x = -0.2$  (ovvero la parabola approssimante centrata in 0 e calcolata in  $-0.2$ ) che risulta essere  $y = 4.72$ .

Considerando quindi nell'intorno di ogni ascissa traslata tre valori della funzione sinistra ricampionata, sono state calcolate le ordinate della relativa funzione traslata, ottenendo una funzione confrontabile, punto per punto, con la funzione relativa al ritaglio destro. In particolare, *l'istogramma combinato dei bordi verticali* vale zero quando le due funzioni dei bordi hanno segni discordi, mentre nel caso di segni concordi è uguale al minimo o al massimo tra le ordinate delle funzioni, a seconda che la  $y$  del vettore sinistro traslato sia, rispettivamente, maggiore o minore di zero. Questo permette di dare maggior peso ai contributi provenienti dal *foreground*, comuni ad entrambe le funzioni, rendendo possibile identificare la presenza di un supporto, anche nel caso in cui i ritagli abbiano uno sfondo rumoroso, caratterizzato da bordi verticali significativi (figura 6.7): grazie al confronto tra le due funzioni le

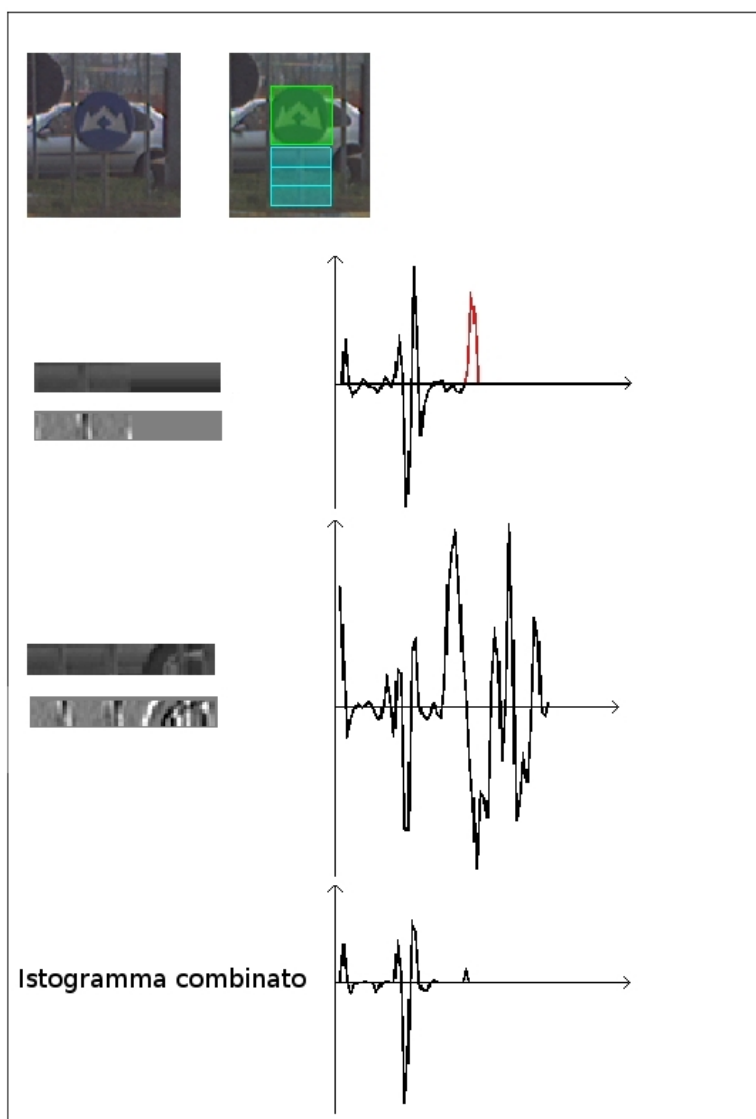


Figura 6.7: Esempio di istogramma combinato, in presenza di sfondo rumoroso. Sono mostrate le funzioni relative al primo terzo dell'area sottostante la *bbox* del cartello. Importante in questo caso è stata l'eliminazione del rumore introdotto dal picco evidenziato in rosso.

componenti di rumore sono infatti sopresse (nel caso di segni discordi) o attenuate (segni concordi).

Nell'istogramma sono portati a zero tutti i valori dell'istogramma combinato minori di una certa soglia, al fine di eliminare i contributi non significativi, e quindi evidenziare l'eventuale presenza di un palo. La soglia introdotta è proporzionale all'altezza del ritaglio e dipende da un parametro impostabile tramite slider.

In alcuni casi però uno sfondo caratterizzato da forti bordi verticali, genera componenti di rumore che, anche se attenuate, presentano picchi maggiori della soglia definita sopra (figura 6.8), anche perché è possibile che un *background* troppo vicino al *foreground* abbia una disparità analoga che porta alla generazione di picchi diffusi. In questo caso comunque si riescono facilmente a distinguere i contributi apportati dai bordi di un palo: questi, infatti, generano picchi più alti rispetto allo sfondo rumoroso. Per questo motivo si è scelto di introdurre un'altra soglia, per portare a zero tutti i valori del vettore di confronto non compresi in un range proporzionale al suo picco. Questa soglia è calcolata in base al picco dell'istogramma combinato dei bordi verticali e ad un parametro impostabile da slider. La sua introduzione non ha comunque permesso di risolvere alcuni casi di bordi nello sfondo particolarmente coerenti.

### 6.3.2 Elaborazione dei bordi orizzontali

L'istogramma dei bordi verticali penalizza alcune tipologie di pannelli integrativi, contenenti in particolare elementi con pochi bordi verticali, come ad esempio le indicazioni di transito per certe tipologie di veicoli come autobus o autocarri (figura 6.9). Per sopperire a questo problema si è scelto di analizzare i bordi orizzontali che caratterizzano i ritagli relativi ai pannelli integrativi. Al fine di ricercare l'eventuale presenza di bordi orizzontali è stato scelto di filtrare le immagini considerando il valore assoluto risultante dal filtraggio; questo perché nel disegno di un elemento con bordi orizzontali, come ad esempio un autocarro, si hanno due variazioni di tono (in corrispondenza del passaggio dallo sfondo bianco alla zona interna del disegno e viceversa) che, per definizione, portano contributi uguali ma con segno opposto al-

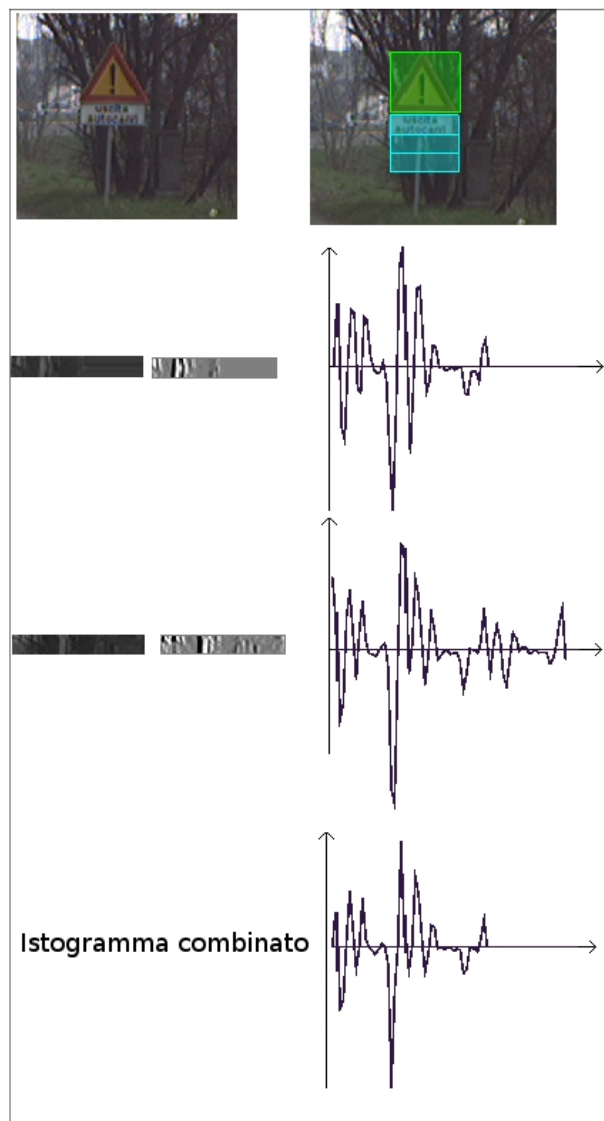


Figura 6.8: Esempio di istogramma combinato, in presenza di sfondo molto rumoroso ad una distanza minima rispetto al cartello: questo comporta, per cartello ed espansioni, valori di disparità molto simili tra loro. Sono mostrate le funzioni relative al secondo terzo dell'area sottostante il *bbox* del cartello.

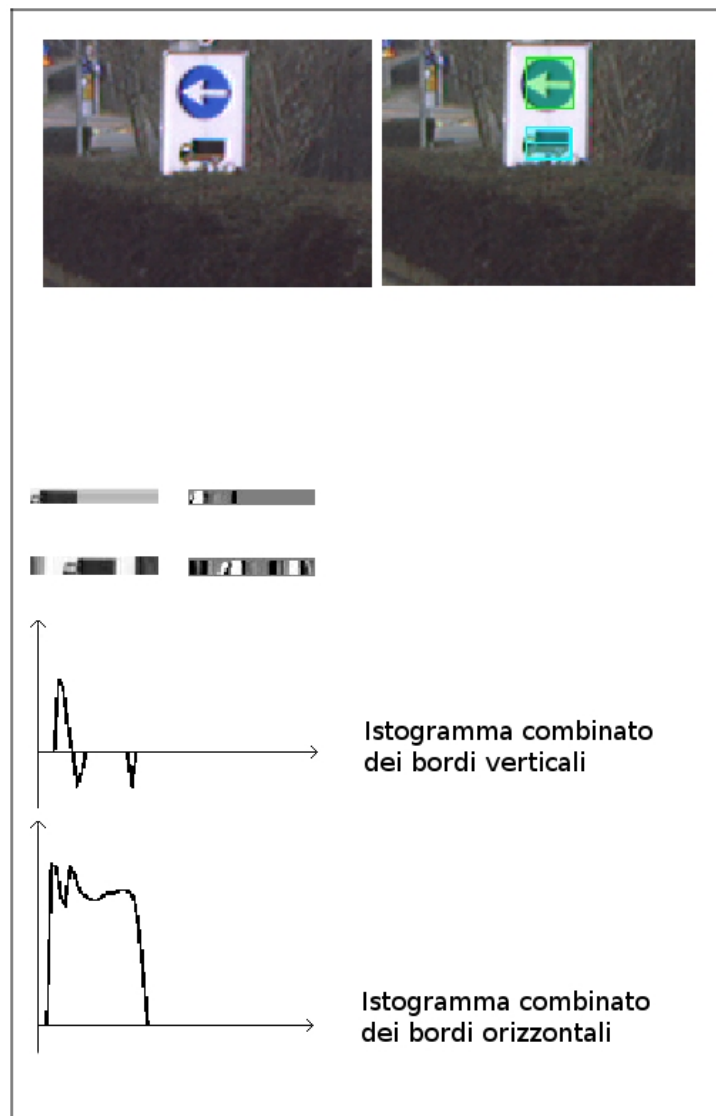


Figura 6.9: Caso di pannello integrativo per cui il calcolo delle funzioni relative ai bordi verticali non era sufficiente. Si calcola quindi l'istogramma combinato relativo ai bordi orizzontali.

l'immagine di *Sobel*. Se si mantenesse quindi l'informazione sul segno, nella somma per colonne, ogni contributo apportato dal primo bordo orizzontale sarebbe annullato dal secondo bordo orizzontale, generando perciò un istogramma nullo.

In particolare sui ritagli filtrati attraverso *Sobel* orizzontale sono state ripetute le medesime operazioni eseguite sulle immagini elaborate con *Sobel* verticale al fine di ottenere, dal confronto degli istogrammi per colonna dei bordi orizzontali provenienti dall'immagine sinistra e destra, *l'istogramma combinato dei bordi orizzontali*.

Per escludere dall'analisi tutti i contributi dovuti a bordi poco significativi, è stata introdotta una soglia, che non dipende dall'altezza del ritaglio perché gli elementi con bordi orizzontali considerati validi portano, nella somma per colonne, lo stesso contributo indipendentemente dalle dimensioni dell'immagine. Infatti se si considera il disegno di un autocarro, al variare della dimensione del ritaglio, rimane costante il numero di bordi orizzontali presenti su ogni colonna (in corrispondenza del passaggio dallo sfondo bianco alla zona interna del disegno e viceversa); questo fa sì che il valore ottenuto dalla somma dei pixel per colonne sia indipendente dall'altezza del ritaglio.

### 6.3.3 Unione degli istogrammi e validazione

Sono quindi presi in esame i due istogrammi combinati, creando un unico istogramma che vale zero quando entrambi valgono zero su cui sono effettuati i controlli per la validazione. Sono state fissate due soglie, una sul numero degli zeri e l'altra sul numero dei valori consecutivi diversi da zero: se il numero dei valori nulli supera la prima soglia o se il numero dei valori consecutivi diversi da zero è minore dell'altra allora il pannello non viene validato. Lo studio di queste funzioni ha quindi permesso di rendere più precisa l'identificazione dei pannelli integrativi: si sono però rilevati casi di falsi positivi causati da un'errata delimitazione del *bbox* del segnale. Infatti qualora il *bbox* sia ritagliato erroneamente, lasciando all'esterno alcune porzioni del cartello, queste rientrano nell'area considerata come espansione; la presenza di bordi orizzontali significativi nella zona preposta ai pannelli integrativi comporta un'errata validazione. Questo problema non è però imputabile al riconoscitore di pannelli in-

tegrativi e deve quindi essere affrontato nel contesto dell'algoritmo di rivelazione dei segnali.

## 6.4 Confronto delle intensità luminose

Le elaborazioni sinora prese in esame non permettono ancora di risolvere alcuni problemi legati:

- alla presenza, a destra di un cartello, di elementi che si trovano alla stessa distanza del segnale dalla telecamera, e quindi sono erroneamente considerati pannelli integrativi destri (figura 6.10);
- alla validazione di espansioni sottostanti relative ad elementi con bordi verticali significativi, associati erroneamente a delle scritte (figura 6.11).



Figura 6.10: Esempio di errata validazione di un'espansione destra: i rami, trovandosi alla stessa distanza del segnale dalla telecamera, generano un valore di disparità prossimo a quello del cartello e quindi sono erroneamente validati.

Allo scopo di ridurre sensibilmente il numero di falsi positivi generati da queste errate validazioni sono state introdotte tecniche di riconoscimento basate sulla valutazione

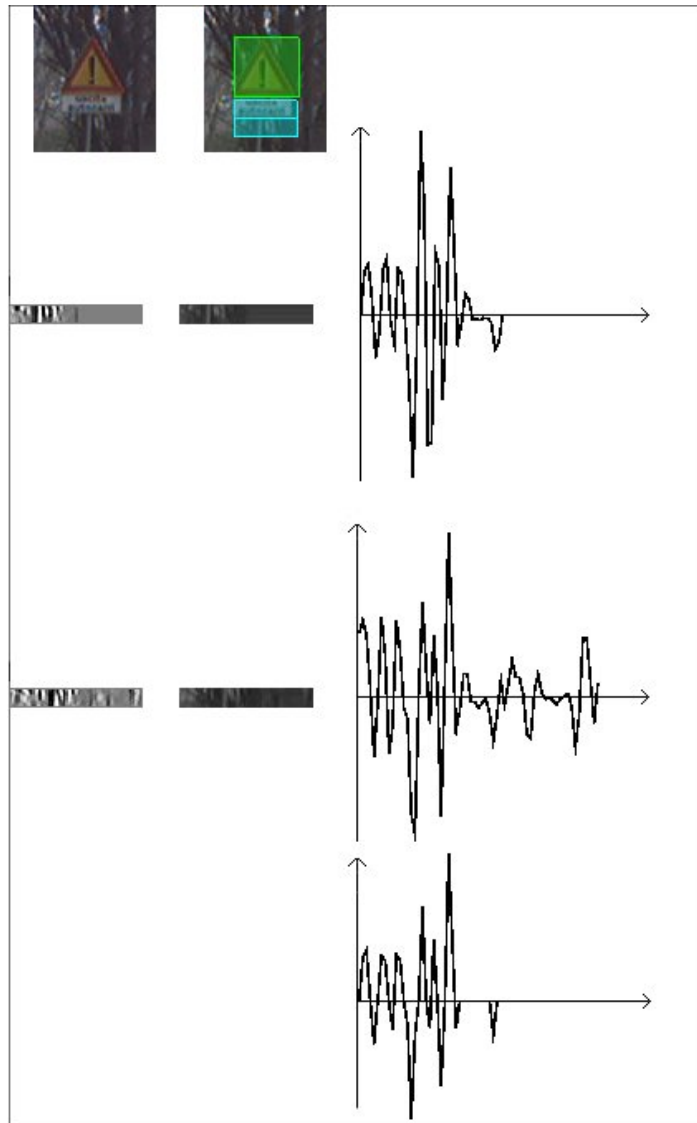


Figura 6.11: Esempio di errata validazione di un terzo sottostante caratterizzato da bordi verticali significativi.

---

e sul confronto dell'intensità luminose nelle regioni prese in esame. In particolare sono considerati validi i pannelli integrativi aventi luminosità compatibile con quella del cartello a cui si riferiscono; per determinare questa compatibilità si confrontano tra loro il valore massimo di luminosità del cartello con quello dell'espansione: se questi due valori sono prossimi tra loro significa che la luminosità del cartello e quella dell'espansione sono coerenti, quindi il pannello integrativo può essere considerato valido.

In particolare, per la valutazione della luminosità di un cartello è calcolata la media, per ogni riga del bounding box che lo contiene, del canale R, G o B considerato dominante per il cartello (l'informazione è contenuta nell'ingresso): l'utilizzo dell'informazione in bianco e nero porterebbe infatti ad una sottostima dell'intensità luminosa. Tra queste medie è scelto il valore più alto, allo scopo di non dipendere da eventuali scritte o disegni (solitamente di colore scuro) eventualmente presenti. Relativamente alle espansioni, il procedimento è analogo, ma la media per riga è calcolata sulle immagini in scala di grigi. Le espansioni possono essere validate se il valore di luminosità calcolato per esse è compatibile con quello calcolato per il cartello.

Un caso particolare si è presentato al momento del calcolo dell'intensità luminosa dei cartelli di forma triangolare: per questa tipologia di segnali, infatti, più ci si allontana dalla base del triangolo, più i valori appartenenti al *background* contenuti nel *bbox* (che è rettangolare) contribuiscono al calcolo delle medie, causando un errore sulla stima della massima luminosità. Per risolvere questo problema si è scelto di considerare, per ogni cartello triangolare individuato, l'immagine contenuta nella metà del *bbox* dal lato della base del triangolo (in cui l'area del *foreground* è prevalente sul quella del *background*) e stimare in questa zona il valore massimo di luminosità, considerando solo i pixel appartenenti al triangolo inscritto nel *bbox* (fig. 6.12 ).

Poiché l'algoritmo, per ciascun cartello, prevede tre possibili espansioni sottostanti (corrispondenti a regioni con altezza pari ad un terzo del *bbox* associato al cartello), si è reso necessario verificare, inoltre, che queste tre zone avessero valori di luminosità prossimi tra loro. Per stimare il valore di luminosità è stata introdotta una funzione, che somma per righe i pixel dell'immagine a toni di grigio associata al ritaglio di

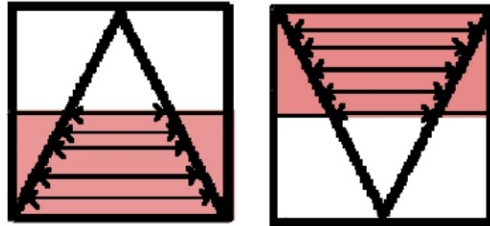


Figura 6.12: Stima del valore massimo di luminosità nel caso di cartelli triangolari. In rosso (grigio) sono evidenziate le righe per le quali il *foreground* è prevalente sul *background*. Le frecce mostrano l'area presa in considerazione al momento del calcolo del valore medio per riga.

ogni terzo sottostante e calcola il massimo tra le somme trovate, ovvero il valore che per l'espansione, presumibilmente, corrisponde al colore bianco. L'algoritmo verifica che il massimo calcolato per gli ultimi due terzi sia compreso in un range di validità, che dipende dal massimo calcolato per il terzo soprastante e da un parametro settabile da slider. Se questa condizione non è soddisfatta significa che l'espansione in questione ha una luminosità diversa rispetto al terzo soprastante e quindi deve essere invalidata.

## 6.5 Tempi computazionali

Come è logico dedurre, i tempi di esecuzione dell'algoritmo sono dipendenti dal numero di cartelli ricevuti in ingresso. Su un sistema Pentium4@2.8GHz è stato rilevato un tempo di elaborazione massimo di 9ms, relativo al frame computazionalmente più pesante tra quelli disponibili, contenente in particolare 3 cartelli.

## 6.6 Risultati e conclusioni

I pannelli integrativi presentano caratteristiche nell'immagine difficilmente prevedibili. In questa trattazione si è cercato di esplorare alcune loro proprietà nel tentativo

di riconoscere il set di caratteristiche che meglio permettano la loro individuazione. In particolare per un pannello integrativo sono state individuate e sfruttate le seguenti proprietà:

- distanza dalla telecamera compatibile con quella del cartello associato (visione stereo);
- posizione e dimensioni prevedibili note le caratteristiche del cartello associato;
- presenza di scritte o disegni, riscontrabile tramite gli istogrammi per colonna dei bordi verticali e dei bordi orizzontali (con utilizzo di visione stereo);
- luminosità compatibile con quella del cartello associato;
- luminosità compatibile per espansione contigue (parti di un pannello integrativo più grandi).

I risultati ottenuti sono buoni (fig. 6.13), ma non ancora soddisfacenti sia in termini di falsi positivi (fig. 6.14), sia in termini di costo del sistema, in quanto la sola individuazione dei pannelli integrativi non può giustificare l'installazione di una seconda telecamera per la realizzazione di un sistema stereo.

Sotto questo punto di vista, risultano molto promettenti i risultati ottenuti attraverso il confronto dell'intensità luminosa del cartello e delle regioni candidate ad essere pannello integrativo. Questo importante risultato dovrà essere sfruttato quando sarà preso in esame lo sviluppo di un algoritmo di ricerca di pannelli integrativi su singola immagine.



Figura 6.13: Esempio di corretta rivelazione di pannelli integrativi.



Figura 6.14: Esempio di un falso positivo, sotto il cartello di divieto d'accesso più in alto, causato dal bordo orizzontale del cartello contenuto nell'espansione ed erroneamente considerato come bordo orizzontale di un possibile disegno contenuto nel pannello integrativo.

# Conclusioni

In questa tesi sono stati presentati diversi sistemi e algoritmi di percezione per uso su autoveicoli basati su visione artificiale. Come conclusione, vorrei sottolineare un principio abbastanza intuitivo, ma che ha trovato riscontro in molto del lavoro realizzato e nei test sperimentali: più è alto il numero di pixel che si riesce a considerare durante i calcoli e più l'informazione estratta sarà affidabile.

Per esempio, con l'uso dell'immagine ternarizzata (app. A) senza necessità di soglie da regolare, è stato possibile ottenere una V-Disparity image di buona qualità anche in ambienti off-road, con la possibilità di estrarre una linea di correlazione affidabile per la stabilizzazione delle immagini. Lo stesso principio è valso nel calcolo della trasformata di Hough per i cerchi, per la quale valutare anche i contributi di punti sui bordi meno contrastati ha permesso di prendere in considerazione un maggior numero di casi.

Sempre guidati da questo principio, porta notevoli miglioramenti nell'affidabilità dei risultati l'utilizzo di informazione raccolta su più frame, anche se il difficile in alcuni casi è proprio decidere quali notizie conservare e come utilizzarle nelle elaborazioni successive. I risultati ottenuti negli algoritmi del calcolo del punto di fuga, della rilevazione di corsie e di cartelli di fine prescrizione, sono solo un esempio dei successi che è possibile ottenere tramite queste tecniche.

## Epilogo

Nel giugno dello scorso anno mi trovavo ad Oshkosh per la Site Visit, l'esame a cui il DARPA sottoponeva il TerraMax2 per l'ammissione alle qualificazioni ufficiali. Mentre il veicolo compiva diligentemente i suoi giri, io e Winfried, stanchi dalle lunghe giornate di preparazione e oppressi dalla calura micidiale di quel giorno, eravamo sotto il tendone della squadra cercando di trovare ristoro. Un giornalista (molto) locale ne stava approfittando per farci qualche domanda.

Parlando dei "suoi" tre laserscanner Ibeo montati sul veicolo, Winfried vantava (con buona ragione) gli oltre 100 mila punti in coordinate mondo al secondo che i sensori raccoglievano. Pensando alle "nostre" 11 telecamere sparse per il veicolo, scherzai dicendo *"Sarebbe bello contare anche quanti pixel raccolgono le telecamere del VisLab!"*.

Ho fatto il conto adesso. Il numero è 140 313 600. Ma mentre dicevo quella frase, sapevo di stare lasciando fuori dal piatto della bilancia una grossa verità: i punti calcolati dal laserscanner sono una informazione raffinata, coordinate (x, y, z) del mondo di notevole precisione, mentre le immagini sono milioni di byte a cui trovare un senso è difficile.

E' stato un successo riuscire a convincere i nostri committenti americani della Oshkosh che la visione poteva dare un contributo per il TerraMax, è stato un successo riuscire a dimostrarlo con il completamento della Grand Challenge, è stato un successo vedere la visione ancora tenere nella sua corsia il TerraMax2, mentre tutti gli altri concorrenti ancora una volta si affidavano a GPS e laserscanner in posizioni improbabili per un veicolo commercializzabile.

Mi chiedo se il mio contributo sia servito a qualcosa, alla creazione un domani di un sistema di assistenza alla guida a basso costo che possa evitare anche un solo incidente mortale. Ovviamente non so la risposta, ma ogni tanto getterò un'occhiata speranzosa agli sviluppi di questa storia.

# **Appendici**



## Appendice A

# V-Disparity Image Analysis

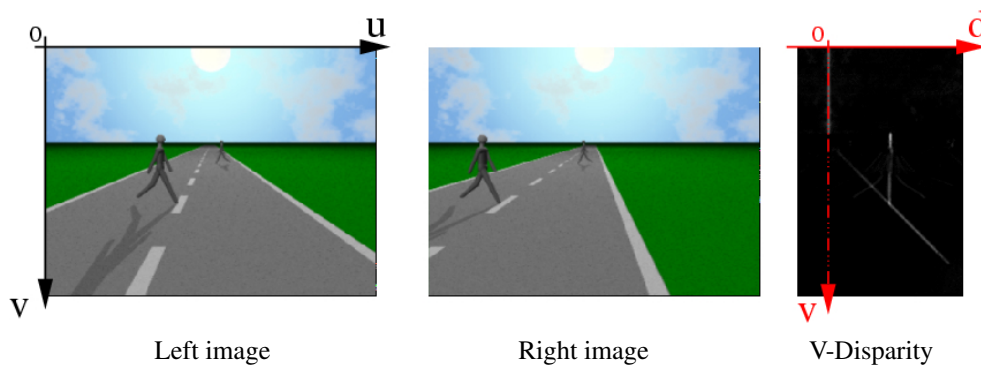


Figure A.1: A V-Disparity image example on synthetic images. The frame of reference is plotted. In this case the similarity measures are computed on edges.

V-Disparity images (Fig. A.1) are 3D graphical representations of the similarity measures between left and right image rows ( $v$  coordinate) depending on the disparities ( $d$  coordinate) used to compare them. Brightness, that is used as third dimension, is directly proportional to the similarity measured. In the example of Fig. A.1, the slanted segment in the V-Disparity image is caused by the linearly changing maximum similarity disparity among ground components. This segment, called *ground correlation*

*line* (briefly *GCL*) in this study, contains the information about the position and the shape of the ground in the images.

## A.1 Computation

The similarity measure of image rows can be computed in many ways. In their works, Labayrade and others [8, 30] compute a fast DSI; then they obtain the similarity collecting, for each image row, the number of matches for each disparity. In [3] we introduced a slightly different approach to V-Disparity image computation, considering comparison windows that correspond to the image lines. That is, similarity is measured among windows 1 pixel high and of the same width of the image. Making this choice it is considered that ground has different disparities at different heights in the image, so it is preferable to maintain windows as short as possible.

In this step the goal is to extract ground information from images. In off-road environments, a good method to do this is to apply a vertical Sobel mask to the image, and to map the obtained values in a ternary domain  $(-1,0,+1)$  obtaining the so-called ternarized image. This approach is thresholdless and takes advantage of a property of the ground that is usually verified: it takes up the largest part of the image. Fig. A.2 shows an example of ternarized images and of the derived V-Disparity image.

Code optimization allows to compute the whole V-disparity image in less than 6 ms on a Pentium IV @ 2.8GHz processor system.

Considering a flat ground, the behavior of the ground correlation line in case of vehicle pitch variations is to oscillate parallel to itself. Knowing cameras static calibration the algorithm can compute a set of candidate ground correlation lines by varying the pitch angle. By collecting the V-Disparity image values along these lines and choosing the best one, it is possible to estimate the pitch at time of acquisition (see Fig. A.3).

The information obtained at this step is used in the subsequent obstacle detection to locate a region of interest in the image where to compute the DSI. It can also be forwarded to other sensors (like laserscanner [31] and vision path detector [32]) in

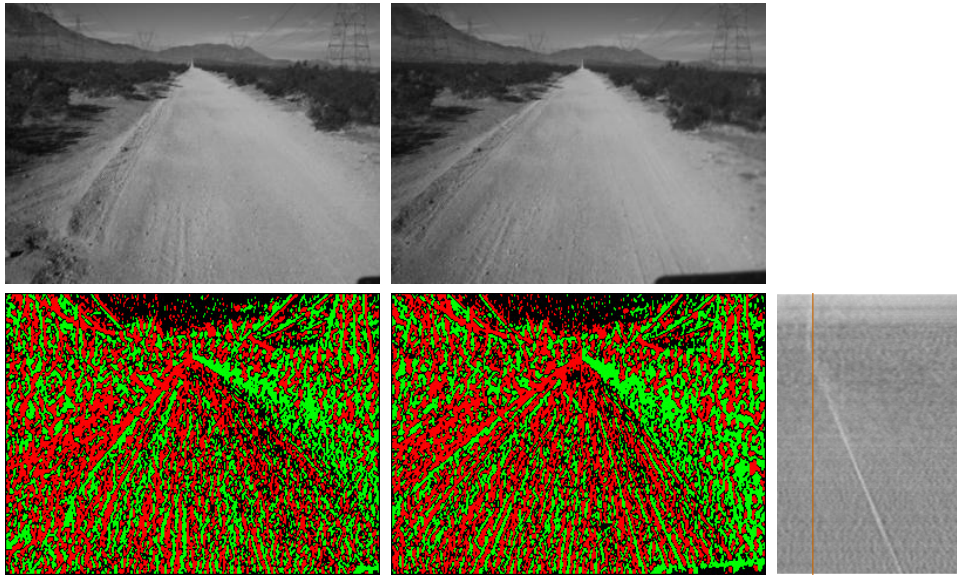


Figure A.2: Original stereo pair, ternarized images, and their V-Disparity image. In the ternarized images, red means negative Sobel phase, whereas green means positive Sobel phase. In the latter, the red line lays on the zero disparity value.

order to improve their environment knowledge. The horizon estimation can also be useful to select an appropriate image area to be used to tune the cameras automatic gain control.

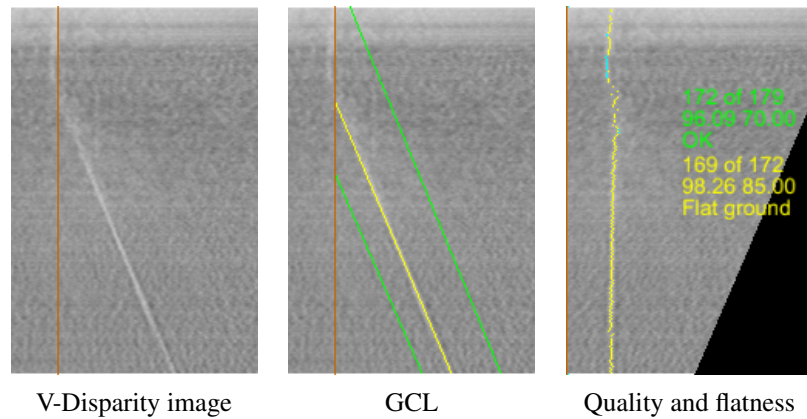


Figure A.3: V-Disparity image analysis. In the center image, the ground correlation line (GCL) is plotted in yellow. Green lines bound the search space. In this case, the ground correlation line is closer to the left bound, so that the horizon height in the image (which corresponds to the point where red and yellow line intersect) is below its position when the vehicle is static. The algorithm can then state that the vehicle is pitching up. In the right image, set A points (see section A.2) are plotted in yellow and set B points in cyan. Green numbers describe V-Disparity image quality analysis. 179 is the number of maxima below the horizon, that are then taken into consideration during the subsequent analysis. 172 is the sum of set A and set B cardinalities. 96.09 is the percentage ratio between 172 and 179 and 70.00 is the threshold applied to distinguish between a good quality image and a bad one. Yellow numbers describe flatness analysis. Again, 172 is the sum of set A and set B cardinalities, while 169 is set A cardinality. 98.26 is the percentage ratio between 169 and 172 and 85.00 is the threshold applied to decide about ground flatness. In this case quality is high and road is flat.

## A.2 Quality and road flatness measurement

A verification of the pitch estimation result would be appreciable because the V-Disparity image quality may be insufficient to reliably detect the ground correlation line. Possible causes of low quality V-Disparity images are underexposed and overexposed images, cameras misalignments, textureless terrains, and strong roll angles. Furthermore, in unstructured environments the flat road assumption is too strong; anyway V-Disparity image contains information about slope changes, since slope changes bend the ground correlation line.

In the following step the positions of the V-Disparity image maxima in each row are considered. A maximum close to the ground correlation line increases the probability of facing a flat ground. Maxima are classified in three categories. Set A is defined as the set of maxima falling within a limited range centered onto the ground correlation line. All the other maxima not belonging to set A, are either rejected or included in set B. In particular, a maxima is rejected when it is isolated (i.e. the number of maxima that can be found in a given small neighborhood is lower than a threshold). To ease the computation of set A and set B, the V-Disparity image is warped [16] using the ground correlation line disparities, so that the points of set A are placed in a vertical strip.

The quality of the V-Disparity image is estimated in inverse proportion to the number of isolated best values. The ground flatness is estimated as  $\frac{|set A|}{|set A|+|set B|}$ , where  $|set A|$  is the cardinality of set A. Figure A.3 shows an example of flat ground, while Figure A.4 shows a non flat ground.

Ternarized images are usually a good method to extract ground information. Nonetheless, in some cases the terrain texture is not enough appreciable (e.g. smooth asphalt roads) and even the use of ternarized images cannot provide sufficient information to the V-Disparity image. Anyway, when the quality of the V-Disparity image is low, it could be recomputed using information other than ternarized images (e.g. road markings extraction in the case of asphalt roads).

V-Disparity image quality analysis allows to estimate also the precision of calibra-



Figure A.4: V-Disparity image analysis in a non flat ground scene. Set A points are plotted in yellow and set B points in cyan.

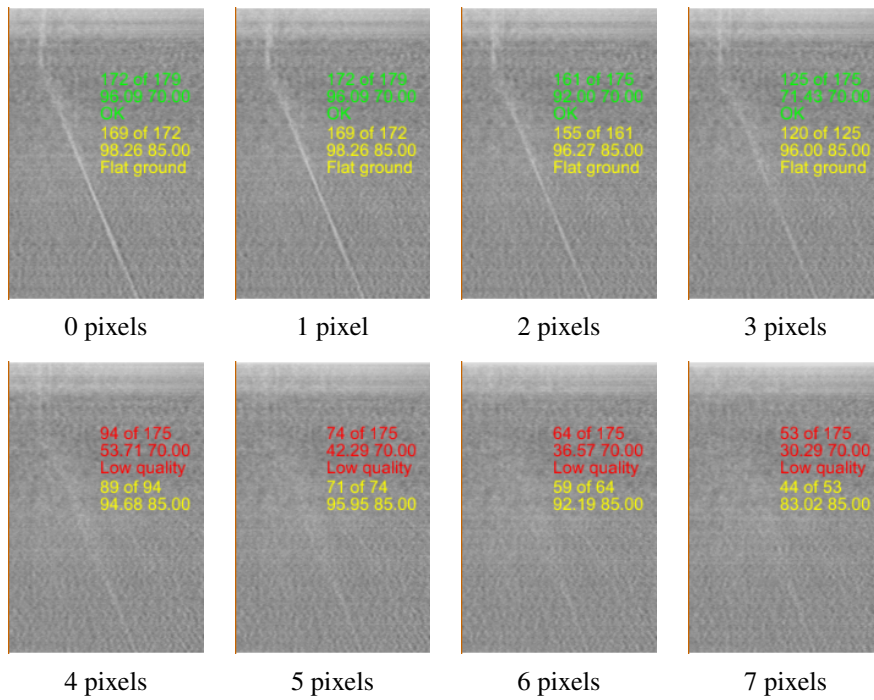


Figure A.5: V-Disparity image quality analysis for different grades of artificial misalignment of images of Fig. A.2. The V-Disparity images computed in correspondence to shifts up to 3 pixels present a sufficient quality to allow a reliable detection of a flat ground.

tion: in particular regarding the cameras alignment. Fig. A.5 shows a sequence of 8 V-Disparity images simulating a growing cameras misalignment, obtained shifting downward the left image of Figure A.2 of 1 pixel steps.

During the processing of an image sequence V-Disparity image quality may change; if this drift remains for a given amount of time, then it can be inferred this is due to a cameras misalignment, caused by a physical movement of the cameras. Although not yet implemented, a software re-alignment (through images shift) can be applied to every incoming image for small angle corrections.

Strong roll angles cause the ground correlation line to appear blurred. Luckily, the TerraMax<sup>TM</sup> vehicle characteristics limited lateral oscillation, so that strong roll angles variations were not registered, so that this problem could be left to further development. A solution could be to perform quality analysis on different V-Disparity images computed on images rotated at different degrees, to discover if the original images were acquired with a non null roll angle. Other approaches [14] are to be considered.

### **A.3 Discussion**

A few cases are not addressed yet. In general the approach fails when the terrain does not feature a prevailing slope, and when the assumption that ground should occupy the largest image part is not met (i.e. in front of wide obstacles). A possible solution to this problem should be to compute the V-Disparity image only considering image regions classified as road by a former algorithm. Anyway, the pitch detection system can detect its own failure in all ground correlation line detection errors through analysis of V-Disparity Image quality.



## Appendice B

# Obstacle Detection

### B.1 Algorithm overview

In the hardware setup of our system the camera pairs have a horizontal baseline. As [15] states, in these conditions, in order to perceive vertical obstacles, it is reasonable to perform localization of homologous points on vertical edges obtained by means of a Sobel filter. Furthermore, a Sobel filtered image does not suffer from stereo matching problems caused by brightness differences between cameras. Unfortunately image areas with constant appearance (low contrast or lack of texture) do not have many (vertical) edges and thus do not provide sufficient information for stereo reconstruction via a simple DSI computation. In such cases, image segmentation can still provide information for stereo reconstruction, but it does so at a high computational cost (e.g. in [33] it takes between 1 and 7 seconds per image). This problem is simply one specific instance of the general phenomena noted in [34]: it is difficult to detect obstacles of any kind with a single approach. The solution to the general problem is to use more than one sensor system. In this specific example, the laser-scanners mounted on TerraMax can easily detect wide textureless obstacles, so the vision system design need not consider obstacles without edges. Exploiting the complementary strengths of vision and laserscanners can provide a complete real-time

sensing solution.

Full 3D stereo reconstruction is currently receiving a great deal of attention by the scientific community; nevertheless, computational time aspects are not always addressed (for example, the algorithm described in [35], that reaches very good performance in terms of 3D reconstruction, takes 40 seconds to process a single pair of images). Obviously, this clashes with our real time requirements.

As a real-time alternative, it is in a two-step approach to full 3D stereo reconstruction. The first step was shown in section A. During V-disparity image analysis, information about position and shape of the ground is collected. This information provides some structure—which off-road environments would otherwise lack. The second step detects obstacles with a DSI-based algorithm that uses information from the first step to reduce the region of interest prior to calculation and to filter the results after calculation. The found obstacles, mapped in real world coordinates, are then grouped following a neighbor rule.

## B.2 DSI computation

Before DSI computation, the algorithm proceeds computing a region of interest using pitch estimation. That is, in order to reduce the computational cost, real world regions too close to the vehicle for a successful stereo matching and image areas well above the estimated horizon are not processed for obstacle detection.

The DSI is computed using a 4 *rows*  $\times$  3 *columns* confrontation window. The chosen window is narrow in order to allow the detection of thin obstacles and short to ease matching of ground features (as stated in section A.1). The DSI is not computed in textureless regions.

The disparity search range is centered considering the expected ground disparity depending on the  $v$  coordinate in the image. Thanks to the code optimization, the DSI is computed within 15 ms with a resolution of  $320 \times 240$  pixels. In the mean-time 3D world coordinates are computed via stereo triangulation, so that the DSI encodes

---

also range information<sup>1</sup>. The wide stereo baselines used here provide a sufficiently accurate depth estimation, thus not requiring sub-pixel accuracy.

### B.3 Obstacles search in the DSI

Many methods have been studied to extract obstacles from a DSI/range image. In road environments, [15] and [14] exploit their knowledge about ground surface in order to delete ground matches in the range image: the remaining points are grouped into obstacles. Anyway, as discussed earlier, the definition of a “ground plane” is not straightforward in off-road environments.

An exhaustive method, suitable for off-road environments, has been introduced in [36] and applied also in [37] and [38]. This approach evaluates connectivity of points in the range image, builds candidate obstacles, and estimates if they present a sufficiently high angle with the ground plane (i.e. they are vertical). A tree search allows then to connect close candidate obstacles and obtain 3D points clouds, whose densities have to be measured in order to classify them as wrong matches or as obstacles. Unfortunately, applications of this method are still below 1.5 Hz throughput. Furthermore, this method seems to be inadequate for thin obstacles detection, one of the main targets of our study.

[39] uses a standard flood filling algorithm to find connected regions (clusters) of similar disparity in the DSI, and evaluates clusters size to decide if they represent obstacles.

In the intent of keeping computational load under control, in this approach they are developed a series of fast filters to be applied to the DSI, in order to detect disparity concentrations that are eligible to be obstacles. A full example of the filter series is shown in Fig. B.4, and is explained step by step in the following lines:

**Ground matches deletion.** If moving up along a DSI column a continuously decreasing disparity is found, this is considered caused by ground elements. Pixels satisfying this property are deleted. Of course, not all the terrain contribution is removed

---

<sup>1</sup>The set of computed 3D world coordinates is called Range Image.

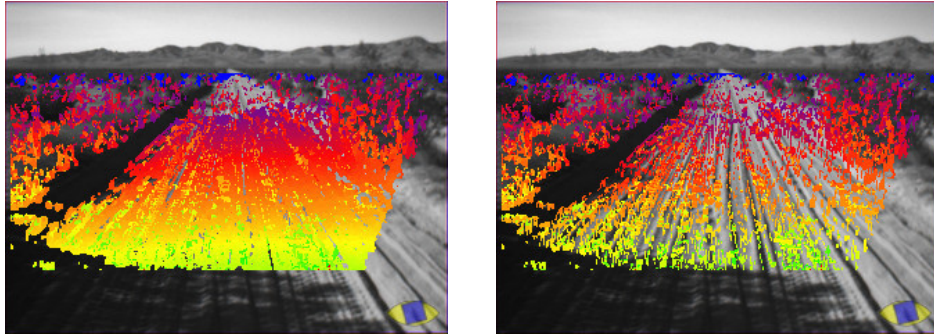


Figure B.1: DSI image before and after ground filter in a highly textured ground situation.

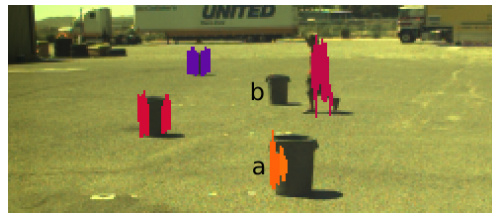


Figure B.2: Example of obstacle detection in case of incorrect column filter behavior. Barrel *b* is not detected because of the interference with the left edge of barrel *a*. The right edge of barrel *a* is not detected because a pedestrian lays in the same columns of the image: even if the pedestrian is farther than the barrel, the pedestrian –being taller– attracts the focus on him.

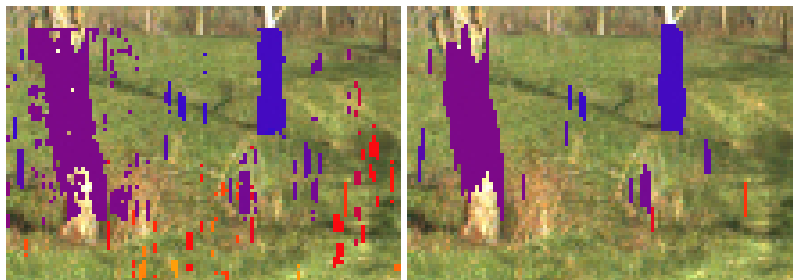


Figure B.3: Processed image of figure B.4 before and after compactness filter (detail). In the second, the speckles are represented centered around their barycenters, and their sizes are proportional to their scores.

at this step because of possible former DSI false matches. Anyway, good results are achieved on highly textured ground regions (Fig. B.1) decreasing the probability of a false obstacle detection, otherwise common in these situations.

**Correlation and distance thresholding.** Poor matchings due to low correlation or relative to too far away areas (low resolution) are then removed (Fig. B.4b).

**Column filter.** Focusing on thin vertical obstacles search, in the next step, only the most common disparity value for each DSI column is preserved (Fig. B.4c), turning off pixels with a different disparity. Even if this step is very effective and draws up significantly the final detection of obstacles, in further development it has to be considered that too much information is lost applying this filter, and other more general approaches have to be contemplated. For example, obstacles at different distances aligned on the same column of the right image are eligible may interfere with each other (see fig B.2). Generally, the farther one (being smaller in the image) is not going to be detected, but it can happen that the obstacle missed is the closer one, or even that both obstacles are missed because of alternate columns prevailing. This system is also unsuitable for the detection of slanted thin obstacles.

The remaining pixel on are now divided in speckles, i.e. the set of pixel still on for each column of the image. These speckles are characterized by a disparity and a 3D world coordinate in the ground plane. This coordinate is obtained by averaging the 3D world  $xy$  coordinates of the points forming the speckle. As mentioned above, these coordinates are contained in the range image obtained during the DSI computation.

**Compactness filter.** The barycenters of the speckles is computed in the DSI. Then, each pixel moves towards the barycenter of the speckle it belongs to; every time it encounters a pixel on (that is part of the speckle) it receives a bonus, while a penalty is given if it has to go through pixels turned off. In this way speckles obtain a score proportional to their size and compactness.

The compactness filter is very effective against false positive caused by repetitive patterns that usually cause discontinuous speckles, formed by (correct) ground matchings and, some lines above, by wrong matches at the same disparity (see fig. B.3).

The drawback is that sometimes obstacles cause discontinuous speckles as well (and consequently get lower scores), especially when the obstacle base is somehow hidden.

**Speckles scoring.** Speckles receive an extra bonus if neighbor columns are ruled by speckles featuring the same disparity. The number of columns in the neighborhood is determined in inverse proportion with the distance characterizing the speckle.

**Thresholding.** Farther obstacles have smaller sizes in the image. To locate obstacles, [15] maps the Range Image data in a bird's eye view map and looks for points concentration. To address the fact that 3D information is more and more sparse as distance grows, it compresses the space proportionally to the distance. Since obstacles are searched directly in the image (and not in a 3D-world representation), the threshold on obstacles size (measured in pixels) are kept in inverse proportion to the distance. Fig. B.4d shows the results at this point of the processing.

**Slope filter.** As fig. B.4d shows, in sloped ground conditions (more precisely, when facing a concave ground), a number of false positives still endures. They are caused by ground regions that present a geometry comparable to obstacle geometry, i.e. an orientation with respect to the cameras similar to a vertical surface resulting in a nearly constant ground disparity on adjacent lines of the images. However, this reflects also on the V-disparity image: in the V-disparity image of the example, letter *a* highlights the nearly constant disparity of ground in the region of images where the ground is oriented almost-perpendicularly with respect to the cameras optical axes. Therefore, as long as the ground occupies a prevailing area in all image rows, the V-disparity image contains information that allow to detect this kind of false positives. For each V-disparity image row a maximum value is found, and the disparity that caused this best match collects one point. After all rows and maxima are considered, it is possible to plot the collected statistics in a histogram (fig. B.5). This histogram is used to raise the applied obstacle threshold specifically on each disparity depending on its score in the histogram. In this way false positives are eliminated, obtaining the final result showed in fig. B.4e.

This approach fails when facing an obstacle that occupies a prominent part of the

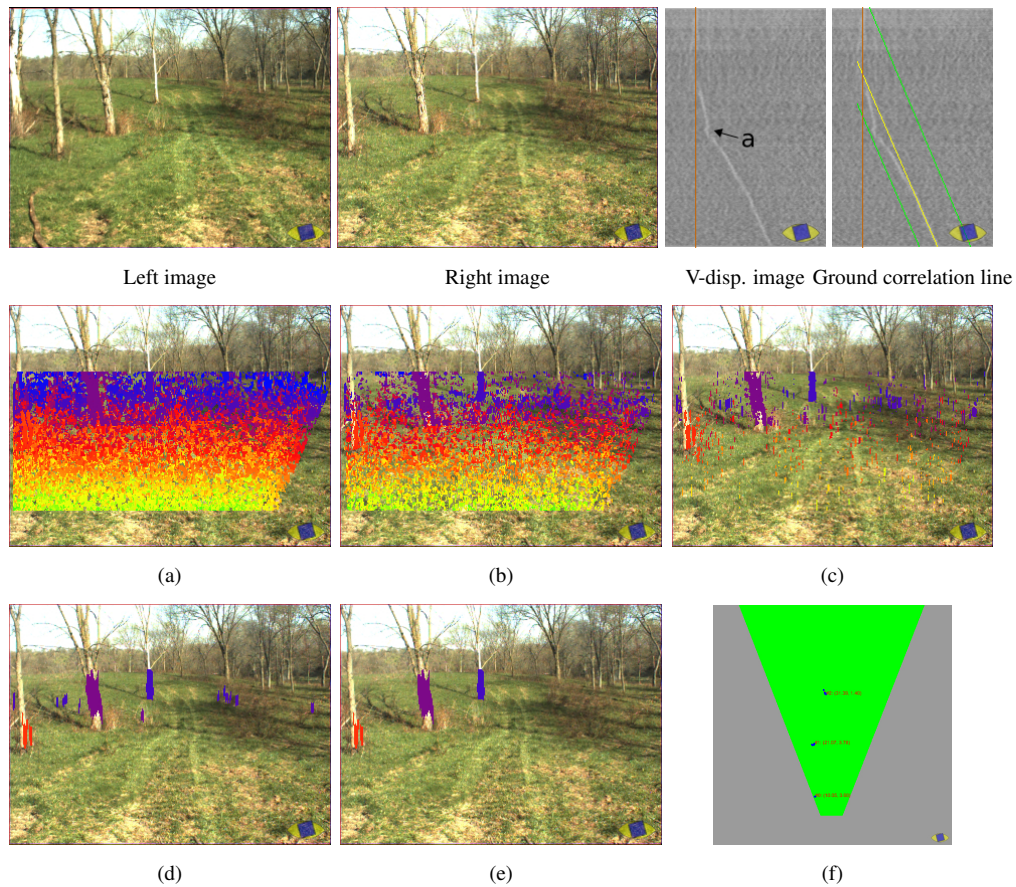


Figure B.4: Example of frame processing in a situation in which the vehicle was facing a short downhill. After the V-disparity analysis, obstacles are found via a series of filters applied to the DSI. (a) *Original DSI.* (b) *After ground, correlation and distance threshold.* (c) *After column filter.* (d) *After compactness and size filter.* (e) *After slope filter.* (f) *Obstacles bird's eye view map.* The slope filter helps to remove the false positives that are mainly present in the region featuring a slope change. Note that in this case the ground filter is not effective because of a discontinuously textured terrain. In the bird's eye view map the field of view is shown in green.

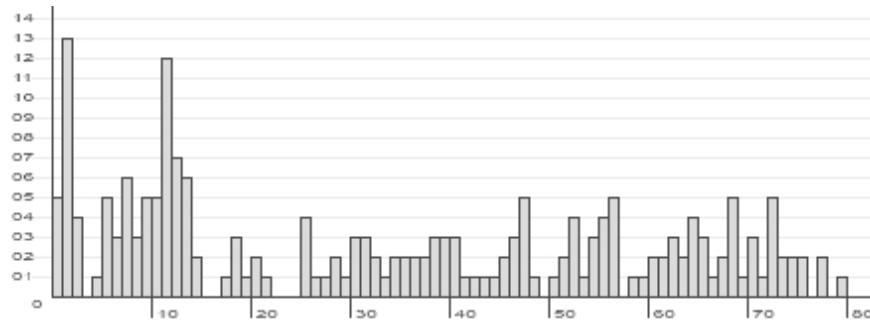


Figure B.5: Number of row maxima caused by each disparity in the V-disparity image of fig.B.4 example. Only rows below the estimated horizon are considered.

image, because it can modify the statistic on V-disparity image masking the ground information. Wide obstacles are in fact considered as slope changes and not detected as obstacle. This problem has not been addressed yet because other sensors mounted on TerraMax can easily detect wide obstacles.

## B.4 Real world coordinates mapping

The obstacles found in the image are not framed with bounding boxes because the algorithm does not need any further step of obstacle classification. As mentioned, the speckles feature a 3D world coordinate, and obstacles are obtained grouping speckles close to each other (e.g. closer than the vehicle width) using a convex hull algorithm. Figure B.4f shows the bird's eye view map of obstacles obtained in fig. B.4e. The whole filtering and real world coordinates mapping are accomplished within 5 ms. Other output examples in different scenarios are shown in Fig. B.6.

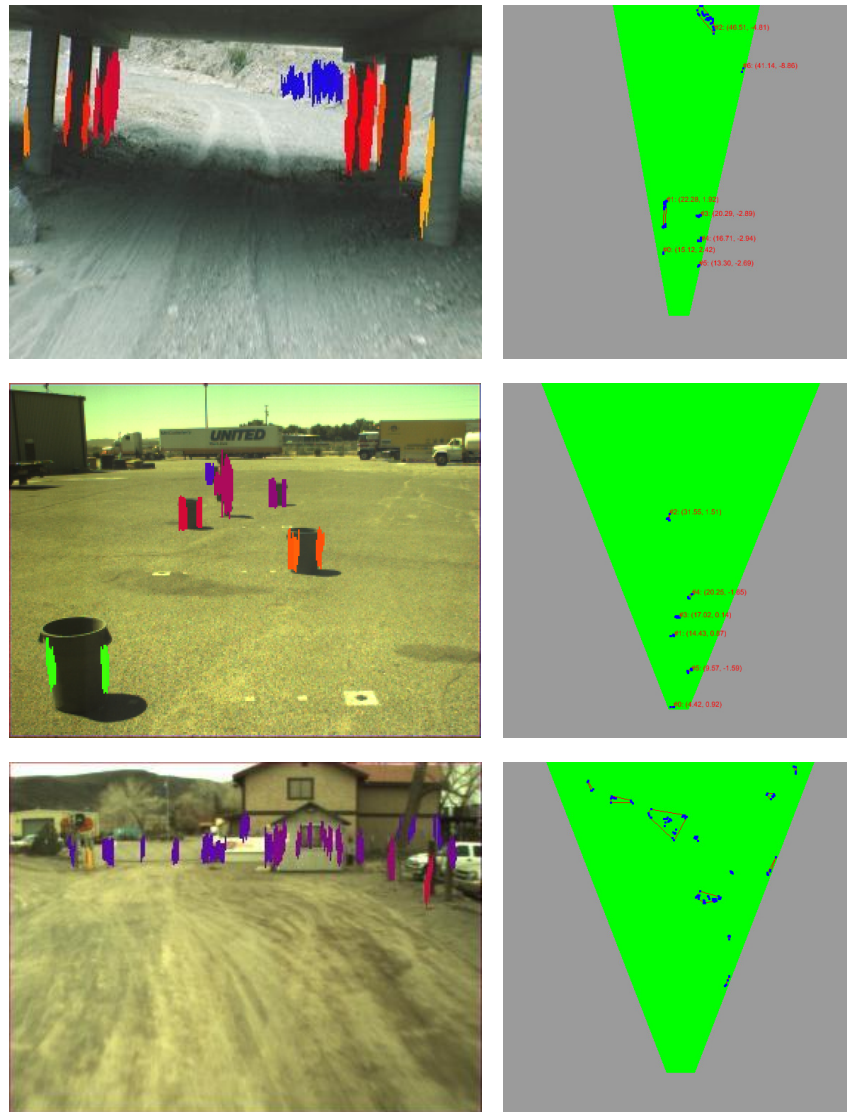


Figure B.6: Obstacle detection for different scenarios. The color of the obstacles varies with distance. In the bird's eye view map the field of view varies because of different baselines and camera setup. In the bottom image, a far gate in the left side is detected (in purple).

## B.5 Obstacle detection results

The algorithm thresholds were tuned in order to avoid false detections. Preparatory testing in the desert near Barstow, California, showed the vision obstacle detection system to be very effectively complemented by the other sensor systems. Vision could detect obstacles which the other sensors missed—namely tall thin ones—and ran with a guaranteed 15 Hz throughput. The obstacle detection system can perform the entire computation within 30 ms.

During the NQE (National Qualification Event prior to DARPA Grand Challenge 2005), the system detected all the bounding cones with only one false detection in four runs. On October, 8<sup>th</sup> and 9<sup>th</sup> 2005 the TerraMax vehicle was one of only five vehicles to complete the entire DARPA Grand Challenge course and was the only finisher exploiting vision for obstacle detection.

Data from the high precision laserscanner helped tune the distance estimates from the vision system to meet all requirements. With accurate calibration, the obstacle detector proved to be robust to false detections, even in complex environments, with slopes, shadows, and different quality of textures. It can also provide features (such as vehicle pitch) to other subsystems, like path detector or higher-level object detectors.

## Appendice C

# Disegno di linee su immagini

Esistono vari approcci per approssimare una linea in una rappresentazione grafica discreta; il più utilizzato è l'algoritmo di Bresenham [40] che risulta essere, oltre che sufficientemente rapido, molto preciso per la realizzazione di questo compito. Di seguito è descritta la procedura per il disegno di una retta generica: nel caso di disegno di un segmento occorre definire gli estremi di inizio e fine della sezione da rappresentare.

Il disegno di una retta avviene attraverso l'incremento dei valori dei pixel da essa attraversata. Si consideri il caso riportato in figura C.1, con una retta di equazione  $y = mx + q$ . La retta attraversa il punto  $P_1(x_1, y_1)$ , che ha coordinate intere, per cui il pixel corrispondente può essere incrementato. Proseguendo verso destra, la retta passa tra i punti di coordinata  $A(x_1 + 1, y_1)$  e  $B(x_1 + 1, y_1 + 1)$ . In questo caso l'algoritmo si comporta distribuendo l'incremento sui due punti, in proporzione alla loro vicinanza.

Questo comportamento è implementato considerando innanzitutto il coefficiente angolare  $m$  della retta  $y = mx + q$  associata ad ogni segmento e distinguendo due casi:

1.  $-1 < m < 1$  (caso in figura);
2.  $m > 1$  o  $m < -1$ .

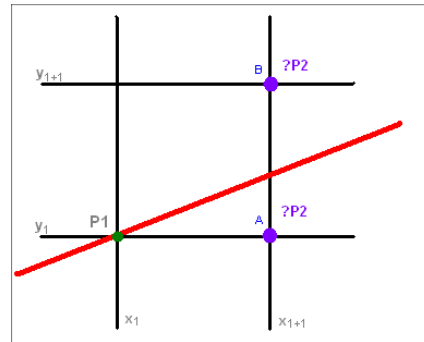


Figura C.1: Rappresentazione di una linea.

Per evitare la generazione di aliasing al momento del disegno, l'algoritmo si comporta in modo diverso nei due casi, scegliendo di utilizzare come ordinata la variabile tra  $x$  e  $y$  che cresce più lentamente:

- Nel primo caso, al variare di  $x$  in  $\mathbb{N}$  da 0 fino alla larghezza dell'immagine di uscita, sono calcolati sulla retta i corrispondenti valori in  $\mathbb{R}$  della  $y$  e incrementati i pixel di coordinata  $y$  a parte intera più vicini. In particolare per un punto  $P(x, y)$  con  $x \in \mathbb{N}$ :

- $P(x, \lfloor y \rfloor)$  è incrementato di  $1 - (y - \lfloor y \rfloor) = \lceil y \rceil - y$ ;
- $P(x, \lceil y \rceil)$  è incrementato di  $1 - (\lceil y \rceil - y) = y - \lfloor y \rfloor$ .

Ad esempio se per  $x = 5$  la  $y$  della retta vale 3.6, allora i valori dei pixel di coordinate  $P(5, 3)$  e  $P(5, 4)$  sono incrementati rispettivamente di 0.4 e 0.6;

- Nel secondo caso invece, al variare di  $y$  in  $\mathbb{N}$  da 0 fino all'altezza dell'immagine di uscita, sono calcolati sulla retta i corrispondenti valori in  $\mathbb{R}$  della  $x$  e incrementati i pixel di coordinata  $x$  a parte intera più vicini. In particolare per un punto  $P(x, y)$  con  $y \in \mathbb{N}$ :

- $P(\lfloor x \rfloor, y)$  è incrementato di  $1 - (x - \lfloor x \rfloor) = \lceil x \rceil - x$ ;
- $P(\lceil x \rceil, y)$  è incrementato di  $1 - (\lceil x \rceil - x) = x - \lfloor x \rfloor$ .

# Bibliografia

- [1] Claudio Caraffi, Stefano Cattani, and Paolo Grisleri. Off-Road Path and Obstacle Detection Using Decision Networks and Stereo Vision. *IEEE Transactions on Intelligent Transportation Systems*, 8(4):607–618, December 2007.
- [2] Alberto Broggi, Claudio Caraffi, Pier Paolo Porta, and Paolo Zani. The Single Frame Stereo Vision System for Reliable Obstacle Detection used during the 2005 Darpa Grand Challenge on TerraMax. In *IEEE Intelligent Transportation System Conference (ITSC06)*, Toronto, Canada, September 2006.
- [3] Alberto Broggi, Claudio Caraffi, Rean Isabella Fedriga, and Paolo Grisleri. Obstacle Detection with Stereo Vision for off-road Vehicle Navigation. In *Intl. IEEE Wks. on Machine Vision for Intelligent Vehicles*, San Diego, USA, June 2005.
- [4] European Road Federation. The ERF's contribution towards enhanced Transport statistics, 2007. <http://www.irfnet.eu/en/2007-statistics>.
- [5] ISTAT. Incidenti stradali, Anno 2006, December 2007. [http://www.istat.it/salastampa/comunicati/non\\_calendario/20071211\\_00/](http://www.istat.it/salastampa/comunicati/non_calendario/20071211_00/).
- [6] PREDIT, Operational Group GO3. *Research and Road Safety. Towards Renew Public Action*. CNRS-MSH-Alpes, November 2006.

- 
- [7] Claudio Caraffi. Stabilizzazione di immagini ed individuazione di ostacoli mediante visione artificiale stereo in ambienti non strutturati. Master's thesis, Facoltà di Ingegneria, Università di Parma, October 2004.
- [8] Raphael Labayrade, Didier Aubert, and Jean-Philippe Tarel. Real time obstacle detection on non flat road geometry through V-disparity representation. In *IEEE Intelligent Vehicles Symposium, Versailles*, pages 646–651, June 2002.
- [9] Alberto Broggi, Claudio Caraffi, Stefano Cattani, and Rean Isabella Fedriga. A Decision Network Based Frame-work for Visual Off-Road Path Detection Problem. In *Procs. IEEE Intl. Conf. on Intelligent Transportation Systems 2006*, pages 951–956, Toronto, Canada, September 2006.
- [10] Marcos Nieto, Luis Salgado, Fernando Jaureguizar, and Julian Cabrera. Stabilization of inverse perspective mapping images based on robust vanishing point estimation. In *Intelligent Vehicles Symposium Proceedings, Istanbul, Turkey*, June 2007.
- [11] S. Nedevschi, F. Oniga, R. Danescu, T. Graf, and R. Schmidt. Increased Accuracy Stereo Approach for 3D Lane Detection. In *IEEE Intelligent Vehicles Symposium*, pages 42–49, Tokyo, Japan, June 2006.
- [12] Massimo Bertozzi and Alberto Broggi. GOLD: a Parallel Real-Time Stereo Vision System for Generic Obstacle and Lane Detection. *IEEE Transactions on Image Processing*, 7(1):62–81, January 1998.
- [13] R. Danescu, S. Sobol, S. Nedevschi, and T. Graf. Stereovision-based side lane and guardrail detection. In *IEEE Intelligent Transportation Systems Conference(ITSC'06)*, pages 1156–1161, Toronto, Canada, September 2006.
- [14] V. Lemonde and M. Devy. Obstacle detection with stereovision. In *Mechatronics & Robotics (MECHROB'04)*, volume 3, pages 919–924, Aachen, Germany, September 2004.

- 
- [15] S. Nedevschi, R. Danescu, D. Frentiu, T. Marita, F. Oniga, C. Pocol, Thorsten Graf, and Rolf Schmidt. High accuracy stereovision approach for obstacle detection on non-planar roads. In *IEEE Intelligent Engineering Systems (INES)*, Cluj Napoca, Romania, 2004.
- [16] Todd A. Williamson. *A High-Performance Stereo Vision System for Obstacle Detection*. PhD thesis, Carnegie Mellon University, September 1998.
- [17] Claudio Caraffi and Stefano Cattani. VisLab at the Grand Challenge. *IEEE Computer*, 39(12):36–37, December 2006. ISSN: 0018-9162.
- [18] Alberto Broggi, Pietro Cerri, Paolo Medici, Pier Paolo Porta, and Guido Ghisio. Real Time Road Signs Recognition. In *Procs. IEEE Intelligent Vehicles Symposium 2007*, pages 981–986, Istanbul, Turkey, June 2007.
- [19] Rossano Vitulli. Riconoscimento e classificazione di segnaletica stradale verticale mediante tecniche di visione artificiale con reti neurali. Master's thesis, Facoltà di Ingegneria, Università di Parma, September 2006.
- [20] Elena Cardarelli. Elaborazione di immagini per individuazione di pannelli integrativi e riconoscimento di segnali di fine prescrizione. Master's thesis, Facoltà di Ingegneria, Università di Parma, December 2007.
- [21] Saturnino Maldonado Bascon, Sergio Lafuente Arroyo, Pedro Gil Jimenez, Hilario Gomez Moreno, and Francisco Lopez Ferreras. Road-sign detection and recognition based on support vector machines. *ITS*, 8(2):264–278, June 2007.
- [22] A. De La Escalera, J.M. Armingol, J.M. Pastor, and F.J. Rodriguez. Visual sign information extraction and identification by deformable models for intelligent vehicles. *ITS*, 5(2):57–68, June 2004.
- [23] Chiung-Yao Fang, Sei-Wang Chen, and Chiou-Shann Fuh. Road-sign detection and tracking. *ITS*, 52(5):1329–1341, September 2003.

- [24] C. Bahlmann, Y. Zhu, Visvanathan Ramesh, M. Pellkofer, and T. Koehler. A system for traffic sign detection, tracking, and recognition using color, shape, and motion information. In *Intelligent Vehicles Symposium (IV2005), Las Vegas, USA*, pages 255–260, June 2005.
- [25] N. Barnes and A. Zelinsky. Real-time radial symmetry for speed sign detection. In *IEEE Intelligent Vehicle Symposium (IV2004), Parma, Italy*, pages 566–571, June 2004.
- [26] S.-S. Ieng, J.-P. Tarel, and R. Labayrade. On the design of a single lane-markings detector regardless the on-board camera’s position. In *IEEE Intelligent Vehicle Symposium (IV’2003)*, pages 564–569, Columbus, OH, USA, 2003. <http://perso.lcpc.fr/tarel.jean-philippe/iv03.html>.
- [27] Gareth Loy and Alexander Zelinsky. Fast radial symmetry for detecting points of interest. *IEEE Trans. Pattern Anal. Mach. Intell.*, 25(8):959–973, August 2003.
- [28] Tilo Burghardt. Circle detection in images using classic sobel filter and hough transformation. Technical Report COMS 30121, University of Bristol, 2003.
- [29] Jose M. Armingol, Arturo de la Escalera, Cristina Hilario, Juan M. Collado, Juan Pablo Carrasco, Marco Javier Flores, Jose Manuel Pastor, and F. Jose Rodriguez. Ivvi: Intelligent vehicle based on visual information. *Robotics and Autonomous Systems*, 55(12):904–916, 2007.
- [30] Raphael Labayrade and Didier Aubert. A single framework for vehicle roll, pitch, yaw estimation and obstacles detection by stereovision. In *Intelligent Vehicles Symposium Proceedings, Columbus*, June 2003.
- [31] Alberto Broggi, Stefano Cattani, Pier Paolo Porta, and Paolo Zani. A Laserscanner–Vision Fusion System Implemented on the TerraMax Autonomous Vehicle. In *IEEE Intl. Conf. on Intelligent Robots and Systems (IROS06)*, Beijing, China, October 2006. In press.

- 
- [32] Alberto Broggi, Claudio Caraffi, Stefano Cattani, and Rean Isabella Fedriga. A Decision Network Based Frame-work for Visual Off-Road Path Detection Problem. In *IEEE Intelligent Transportation System Conference (ITSC06)*, Toronto, Canada, September 2006.
- [33] F.J. Estrada and A.D. Jepson. Quantitative evaluation of a novel image segmentation algorithm. In *CVPR*, volume 2, pages 1132 – 1139, San Diego, USA, June 2005.
- [34] Rankin Arturo, Andreas Huertas, and Larry Matthies. Evaluation of stereo vision obstacle detection algorithms for off-road autonomous navigation. In *32nd AUVSI Symposium on Unmanned Systems*, June 2005.
- [35] J. Sun, Y. Li, S.-B. Kang, and H.-Y. Shum. Symmetric stereo matching for occlusion handling. In *CVPR*, volume 2, pages 399–406, San Diego, USA, June 2005.
- [36] A. Talukder, R. Manduchi, L. Matthies, and A. Rankin. Fast and reliable obstacle detection and segmentation for cross country navigation. In *IEEE Intelligent Vehicles Symposium*, Versailles, France, 2002.
- [37] R. Manduchi, A. Castano, A. Talukder, and L. Matthies. Obstacle detection and terrain classification for autonomous off-road navigation. *Autonomous Robot*, 18, pages 81–102, 2003.
- [38] J.C. van den Hauvel, J.C.M. Kleijweg, W. van der Mark, M. Lievers, and L.J.H.M. Kester. Obstacle detection for people movers using vision and radar. In *10 th World Conference on Intelligent Transport Systems and Services*, Madrid, Spain, 2003.
- [39] Thao Dang and Christian Hoffmann. Fast object hypotheses generation using 3D position and 3D motion. In *Intl. IEEE Wks. on Machine Vision for Intelligent Vehicles*, San Diego, USA, June 2005.

- 
- [40] Jack Bresenham. Algorithm for computer control of a digital plotter. *IBM Systems Journal*, 4(1):25–30, 1965.
- [41] Alberto Broggi, Massimo Bertozzi, Claudio Caraffi, Mirko Felisa, Guido Vezioni, and Mike Del Rose. Pedestrian Detection by means of Far-infrared Stereo Vision. *Computer Vision and Image Understanding*, 106(2):194–204, 2007.
- [42] Massimo Bertozzi, Alberto Broggi, Luca Bombini, Claudio Caraffi, Stefano Cattani, Pietro Cerri, Alessandra Fascioli, Mirko Felisa, Rean Isabella Fedriga, Stefano Ghidoni, Paolo Grisleri, Paolo Medici, Marco Paterlini, Pier Paolo Porta, Matteo Posterli, and Paolo Zani. Vision Technologies for Intelligent Vehicles. In *11<sup>th</sup> Intl. Conf. on Knowledge-Based and Intelligent Information & Engineering Systems*, pages 704–711, Vietri sul Mare, Italy, September 2007.