

UNIVERSITÀ DEGLI STUDI DI PARMA

Dottorato di Ricerca in Tecnologie dell'Informazione

XXV Ciclo

**SVILUPPO DI UNA ARCHITETTURA SOFTWARE
PER VEICOLI INTELLIGENTI BASATA SU
DIVERSI LIVELLI DI AUTOMAZIONE**

Coordinatore:

Chiar.mo Prof. Marco Locatelli

Tutor:

Chiar.mo Prof. Alberto Broggi

Dottorando: *Stefano Debattisti*

Gennaio 2013

*Alla mia famiglia,
e a Monica*

Sommario

Introduzione	1
1 Dalla guida analogica alla guida digitale	7
1.1 Introduzione	7
1.2 Analisi dei possibili livelli di automazione	9
1.3 Gestione dei conflitti	12
1.3.1 Descrizione delle Interfacce uomo macchina	16
1.3.2 HMI per diversi livelli di automazione	20
1.4 Guida a livello di manovra	21
1.4.1 Conduct-by-Wire	24
1.4.2 H-Metaphor	28
2 Contributo alla guida a livello di manovra	33
2.1 Digital Driving	33
2.2 Catalogo delle manovre	36
2.2.1 LKS+ACC	42
2.2.2 Sorpasso	44
2.2.3 Ingresso/Uscita	47
2.3 Behaviours	50
2.4 Risoluzione dei conflitti	57
3 Architettura software basata su diversi livelli di automazione	59
3.1 Open intelligent Architecture for Future Autonomous Vehicle	59

3.2	Stato dell'arte	61
3.2.1	Architettura del team Oshkosh	62
3.2.2	Architettura del prototipo Junior	64
3.3	Architettura software basata su diversi livelli di automazione	66
3.3.1	Vislab Bus	70
3.3.2	World Perception Server	72
3.3.3	Layer di Pianificazione	76
3.3.4	Layer di Percezione	81
4	Moduli di percezione	83
4.1	Rilevamento della superficie stradale basata sul colore	83
4.1.1	Analisi degli spazi di colore CIELab e HSV	84
4.1.2	Algoritmo per il riconoscimento della superficie stradale	88
4.1.3	Risultati	89
4.2	Rilevamento e tracking di un veicolo Leader	90
4.2.1	Immagini Intrinseche	91
4.2.2	Algoritmo	97
4.2.3	Risultati	100
5	Conclusioni	103
A	Considerazioni legali sui sistemi di guida automatica	109
A.1	Leggi regolatrici per i veicoli odierni	109
A.2	Sistemi ad elevata automazione	113
A.3	Sistemi completamente autonomi	114
B	Testing di un veicolo intelligente	115
	Bibliografia	121

Elenco delle figure

1	Benefici dell'utilizzo di un veicolo intelligente.	2
1.1	Tassonomia nell'automazione di un veicolo.	9
1.2	Possibili livelli di automazione in un veicolo intelligente.	10
1.3	Possibili conflitti nei diversi livelli di automazione.	13
1.4	Conflitto durante la guida altamente automatizzata.	14
1.5	Conflitto durante la guida totalmente automatizzata.	15
1.6	Architettura di sistema di Conduct-by-Wire.	25
1.7	Esempio di manovra in CbW: cambio corsia destro.	26
1.8	HMI in Conduct-by-Wire.	27
1.9	Modello di interazione fra due agenti intelligenti.	29
1.10	Connessione guidatore cavallo nel sistema con carretto.	30
2.1	Veicolo convenzionale: divisione in layer.	34
2.2	Veicolo con Digital Driving: divisione in layer.	35
2.3	Percorso del test VIAC.	37
2.4	Struttura della lista di gates per la manovra "LKS+ACC".	43
2.5	Attivazione della manovra di sorpasso.	45
2.6	Sorpasso: lista dei gates.	46
2.7	Ingresso/Uscita: lista dei gates.	49
2.8	Rapporto fra behaviours e manovre nel Digital Driving.	51
2.9	Meccanismo di abilitazione dei behaviours e delle manovre.	52
2.10	Utilizzo della priorità in <i>Highway Mode</i>	57

3.1	Architettura software del veicolo Terramax.	63
3.2	Architettura software del veicolo Junior.	65
3.3	Architettura software basata su diversi livelli di automazione.	69
3.4	Schema a blocchi del WPS.	73
3.5	Composizione del blocco di fusione sensoria Lane+Obstacles.	74
3.6	Approccio gerarchico alla pianificazione, un esempio.	77
3.7	Road Junction Map: compressione degli archi.	78
3.8	Road Junction Map: risultati.	79
3.9	A* planning: risultati.	81
4.1	Spazio di colore CIELab: rapporto a/b al variare di L.	85
4.2	Sfere rappresentanti gli spazi di colore HSV e HSL.	87
4.3	Diagramma rappresentante l'algoritmo di rilevazione della superficie stradale.	88
4.4	Analisi spazi di colore: risultati.	89
4.5	Road detector: risultati	90
4.6	Risposta caratteristica di un sensore narrow-band.	92
4.7	Intrinsic Images: processo di calcolo in due fasi.	93
4.8	Intrinsic Images: meccanismo di funzionamento e asse l_θ	94
4.9	Calibrazione per Intrinsic Images: idea base	95
4.10	Esempio di funzionamento di SURF: invarianza alle trasformazioni geometriche e fotometriche.	98
4.11	Addestramento del modello di colore.	99
4.12	Inseguimento veicolo: applicazione a video VIAC.	101
4.13	Inseguimento veicolo: applicazione a situazioni simulate.	101
5.1	Percorso della dimostrazione finale del progetto OFAV.	106
B.1	Passi fondamentali nell'implementazione di una simulazione in Pre-scan (image credit www.tass-safe.com).	116
B.2	Testing dell'architettura: integrazione del simulatore.	118

Elenco delle tabelle

1.1	Differenze fra DAS e ADAS.	23
1.2	Confronto fra diverse modalità di guida automatizzata.	32
2.1	Manovre derivanti dall'analisi dei dati dal progetto VIAC.	42
2.2	Composizione dei behaviours, il numero in tabella corrisponde alla priorità che la manovra ha all'interno del behaviour.	54
4.1	Risultati con diverse situazioni metereologiche dell'algoritmo di in- seguimento veicolo.	102

Introduzione

I sistemi di trasporto sono da sempre un ambito di primaria importanza all'interno della società, sia che si tratti del trasporto delle persone sia che si tratti del trasporto di beni. Attualmente l'obiettivo della ricerca è assicurare la sostenibilità economica, sociale ed ambientale dei sistemi di trasporto. Uno dei problemi principali per quanto riguarda la sostenibilità sono gli incidenti stradali: nella sola Unione Europea si verificano più di 40000 morti e 1000000 feriti all'anno a seguito di incidenti stradali [1]. In aggiunta, questi incidenti portano un costo annuale stimato di circa 200 miliardi di euro, che equivalgono al 2% del Prodotto Interno Lordo (PIL) dell'intera Unione Europea. Analizzando le cause di questi incidenti si scopre che il 95% di questi incidenti è dovuto a fattori umani [2] [3], in particolare ad errori dovuti a distrazione, sonnolenza od erronea valutazione delle distanze. Il rimanente 5% è dovuto nell'ordine a problemi nelle infrastrutture, al tempo atmosferico oppure a problemi tecnici. Un altro dei grandi problemi legato ai sistemi di trasporto sono le congestioni del traffico e l'inquinamento che ne deriva: si stima che il costo dovuto all'inquinamento rappresenti all'incirca il 2% del PIL dell'intera zona EU e, cosa ancora più importante, si stima che sia una concausa nella morte di 30000 cittadini della zona EU ogni anno [4].

Una soluzione al problema dell'inquinamento potrebbe essere rappresentata dal trasporto di massa (autobus, treni, navi etc.): esso infatti rappresenta un'alternativa sicura e pulita al trasporto individuale (auto e camion). Inoltre, permette ai passeggeri di occupare il tempo di viaggio lavorando o divertendosi senza doversi occupare della guida del mezzo di trasporto. Nonostante ciò, il trasporto individuale è spesso

preferito a quello di massa, infatti il 72,4% dei km/passeggero sono percorsi in auto e per il trasporto di merci sulla terraferma circa il 72.5% delle tonnellate/km sono percorse da camion [4]. Il problema principale del trasporto di massa è la poca efficienza

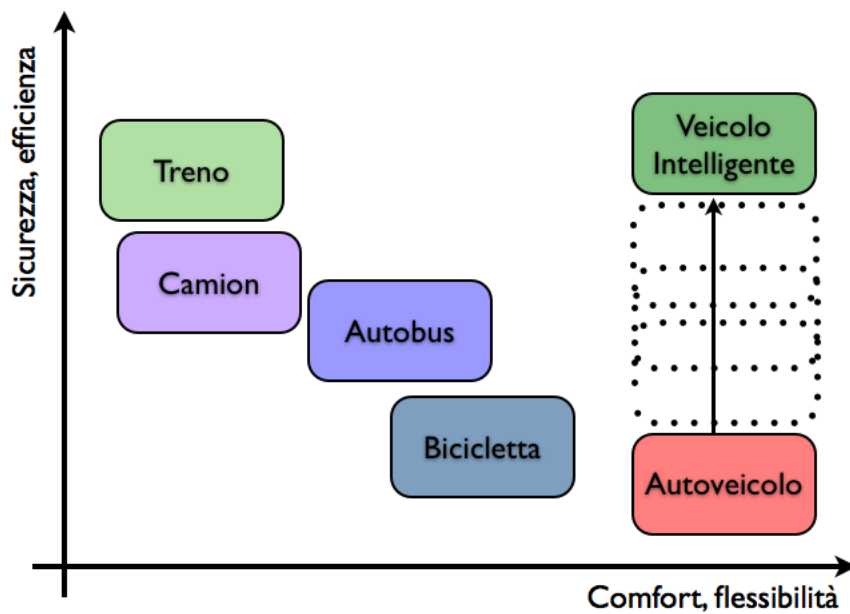


Figura 1: Benefici dell'utilizzo di un veicolo intelligente.

dal punto di vista del tempo di viaggio: infatti normalmente il trasporto di massa è effettuato solamente rispettando una certa tabella oraria, inoltre non porta esattamente da un punto A ad un punto B ma bensì porta da un punto C ad un punto D, lasciando all'utente il compito di spostarsi da A a C e da D a B. Inoltre effettua fermate regolari lungo il tragitto, perdendo altro tempo. Tutto questo è particolarmente vero per tratte che non riguardano le grandi città, le quali normalmente sono meglio servite rispetto alle zone poco densamente abitate. Il trasporto individuale è normalmente preferito poiché permette di evitare tutte le perdite di tempo dovuto alle tabelle orarie e alle fermate lungo il percorso e permette inoltre un flusso di traffico molto maggiore nella

stessa area infrastrutturale.

Una soluzione ai problemi visti in precedenza potrebbe essere rappresentata dai veicoli intelligenti, poiché potrebbero aumentare sicurezza ed efficienza dei trasporti. In particolare l'utilizzo dei veicoli intelligenti potrebbe combinare i vantaggi dei veicoli attuali, e quindi i vantaggi in generale del trasporto individuale, con i vantaggi del trasporto di massa, come raffigurato in figura 1.

I veicoli intelligenti potranno offrire la flessibilità del trasporto individuale, mantenendo un elevato tasso di sicurezza garantita dal fatto che il sistema di guida automatica non sarebbe soggetto a fatica, sonnolenza o a particolari stati emotivi tipici di un conducente umano. L'efficienza temporale sarebbe massima poiché essi potrebbero effettuare il trasposto tra A e B 24/7, inoltre avrebbero il vantaggio di essere accessibili a persone di tutte le età, senza richiedere particolari abilità, inoltre renderebbe il trasporto accessibile anche ai disabili, sia permanenti che temporanei. I tempi di reazione molto brevi tipici dei sistemi computerizzati permetteranno inoltre la creazione di convogli a colonna, come per esempio treni costituiti da singoli veicoli, raggiungendo in questo modo l'efficienza del trasporto di massa mantenendo il comfort del trasporto individuale, come già raffigurato in figura 1.

Molti sono i centri di ricerca attivi nel campo dei veicoli intelligenti, e più in generale dell'*Intelligent Transportation Systems* (ITS); per contro i produttori di auto sono molto cauti poiché è un tipo di ricerca a lungo termine e ad alto rischio. In particolare, per quanto le tecnologie ricavate da questo tipo di ricerca siano valutate come interessanti da tutti i soggetti interessati (anche nel breve-medio periodo) lo sviluppo di un veicolo intelligente è considerata un tipo di ricerca ad alto rischio, soprattutto a causa delle difficoltà di carattere legale e di carattere tecnologico che questa attività implica. Dal punto di vista industriale ed economico grandi ritorni si potranno avere una volta accertata la maturazione della tecnologia, la risoluzione dei problemi legali e l'accettazione da parte degli utenti di questi tipi di aiuti alla guida. La base per gli investimenti da parte dei produttori di auto in questo campo potrà venire dall'esperienza maturata da diversi laboratori di ricerca, i quali stanno lavorando nel campo dell'ITS da diversi anni. In particolare, eventi come le DARPA Grand e Urban Challenges [5] [6] hanno dimostrato il grande potenziale dei veicoli intelligenti; in ogni

caso i prototipi sviluppati in occasione di questi eventi non risultano ancora pronti per l'utilizzo nelle situazioni di traffico di tutti i giorni. Infatti, prendendo ad esempio la Urban Challenge, si aveva una città chiusa al pubblico, con solo pochi veicoli non robotici guidati da *stuntman* professionisti molto attenti al rispetto di ogni regola stradale, e non erano presenti semafori, pedoni o biciclette; in pratica si aveva un ambiente il più possibile amichevole per i partecipanti alla gara. Ne consegue che pur rappresentando un grande evento nel campo dell'ITS i veicoli risultanti da questo tipo di competizioni non sono pronti per essere immessi sul mercato poiché sono stati progettati per partecipare a dimostrazioni con regole ben precise e di conseguenza risultano funzionanti solamente sotto quelle determinate condizioni. Lo stesso discorso vale per i prototipi usciti da altri progetti come il progetto europeo IP PReVENT, o come il progetto europeo Interactive [7] [8] le cui dimostrazioni si sono tenute in circuiti chiusi o in ambienti fortemente strutturati come ad esempio autostrade. Un'ultima problematica riguardante questo tipo di prototipi riguarda il fatto che essi sono dotati di una sensoristica che per dimensioni e costo non può essere utilizzata su veicoli di serie; inoltre la particolare collocazione dei sensori (sul tetto, dietro il parabrezza o in posizione laterale) di fatto ne diminuisce la sicurezza limitandone la visuale, ne diminuisce l'aerodinamicità e ne peggiora decisamente l'aspetto esteriore, diminuendo quindi le possibilità di accettazione da parte dell'utente di questo tipo di sistemi.

Dal punto di vista funzionale gli eventi appena citati hanno evidenziato che i punti critici di questi sistemi sono rappresentati dai sistemi di percezione e pianificazione delle azioni, i quali devono tener conto di situazioni reali che possono essere molto complicate e di difficile comprensione. Infatti i sistemi di controllo utilizzati durante la Urban Challenge furono spesso derivati direttamente da quelli creati per la Grand Challenge, modificandoli e adattandoli semplicemente all'ambiente urbano, poiché il problema del controllo di un veicolo è intrinsecamente più strutturato e meglio definito rispetto al problema della percezione o della decisione del comportamento che il veicolo dovrà tenere. Il problema della percezione è molto complesso e spesso non ben definito: un sistema di percezione ha infatti a che fare con situazioni molto differenti e spesso molto complesse in cui gli oggetti sulla scena devono essere per-

cepiti e correttamente classificati; la presenza di molti oggetti sulla scena stessa può portare ad errori nella percezione e di conseguenza introdurre rumore nel risultato finale. Considerazioni simili possono essere fatte per la parte di decisione: per direzionare correttamente il veicolo non è sufficiente conoscere esattamente il modello del veicolo e saperlo controllare alla perfezione, ma è necessario anche modellare il mondo esterno, tenendo in considerazione un gran numero di diverse situazioni. Per questo motivo lo sviluppo di sistemi di percezione e decisione necessitano, allo stato attuale delle cose, una ricerca maggiore rispetto ai sistemi di controllo di un veicolo; inoltre data la difficoltà del compito può essere presa in considerazione l'introduzione della cooperazione fra veicoli e conducente al fine di discriminare situazioni particolarmente complesse.

In questa tesi è presentato il lavoro di ricerca che ho svolto durante i 3 anni di dottorato di ricerca presso il Dipartimento di Ingegneria dell'Informazione dell'Università degli Studi di Parma; il mio lavoro è incentrato principalmente sui veicoli intelligenti ed in particolare sullo sviluppo di architetture software per veicoli intelligenti che siano in grado di supportare diversi livelli di automazione integrando allo stesso tempo la cooperazione fra conducente e veicolo intelligente; nel capitolo 1 è presentata un'analisi dei vari livelli di automazione all'interno di un veicolo e un'analisi dei possibili conflitti fra guidatore e veicolo, nel capitolo 2 è presentata la proposta di un nuovo paradigma di guida altamente automatizzata. Nel capitolo 3 è presentata un'architettura per veicoli intelligenti basata su diversi livelli di automazione, con una descrizione dei principali componenti sviluppati mentre nel capitolo 4 vengono analizzati le componenti di percezione interne all'architettura stessa. Infine nel capitolo 5 vengono presentati i risultati e i prossimi sviluppi dell'attività di ricerca svolta durante questi anni di dottorato.

Capitolo 1

Dalla guida analogica alla guida digitale

*L'informatica non riguarda i computer più di
quanto l'astronomia non riguardi i telescopi*

Edsger Dijkstra

1.1 Introduzione

Attualmente i due temi di ricerca principali incentrati sui veicoli intelligenti sono gli *Advanced Driver Assistance Systems* (ADAS) e i veicoli autonomi. L'interesse nella ricerca nel campo dei veicoli autonomi è stato elevato sin dalla fine degli anni '90, quando il raggiungimento dell'obiettivo della produzione di un veicolo completamente autonomo sembrava essere molto vicina; allo stesso tempo gli ADAS iniziavano ad attrarre fortemente le aziende produttrici di veicoli, in particolar modo poiché la produzione in serie era quasi attuabile con costi limitati. Negli ultimi anni, grazie alla spinta di alcune aziende leader nei rispettivi settori e grazie ai nuovi bisogni imposti dalla Green Economy l'interesse per i veicoli autonomi è tornato ad essere molto alto

in particolare in alcuni campi ben definiti come quello agricolo, minerario, nell' esplorazione ed, infine, in quello militare. Gli ADAS sono progettati prevalentemente per essere utilizzati in scenari predefiniti e molto circoscritti, come ad esempio nelle autostrade, nei parcheggi, nelle situazioni di guida in coda e così via; le loro performance devono essere molto elevate nei rispettivi domini di applicabilità con basse tolleranze d'errore. Oltre a questo, costi e dimensioni ridotte sono caratteristiche essenziali che ne permettono una facile integrazione ed una conseguente commercializzazione nel mercato dell'automobile [9, 10]. Per contro, nei veicoli autonomi il fattore costo e il fattore dimensione sono meno importanti in quanto i sistemi creati non devono essere integrati in veicoli preesistenti ma rappresentano un gamma di autoveicoli completamente nuova; d'altro canto i veicoli autonomi devono far fronte a situazioni molto più varie, meno strutturate ed in particolare devono far fronte a tutte le condizioni atmosferiche, tutti i tipi di terreni e tutti gli scenari che un normale veicolo si trova ad affrontare durante il suo ciclo di vita [11, 12, 13].

Nonostante le differenze appena evidenziate fra i due filoni principali di ricerca, molte delle idee base sono comuni; ad esempio i concetti di percezione, pianificazione e attuazione sono identici in entrambi i sistemi ma i differenti requisiti in termini di costo e integrazione portano a sistemi differenti di percezione e processing dei dati acquisiti [14]. In estrema sintesi la differenza principale fra i due tipi di sistemi risiede nella presenza o nell'assenza del guidatore a bordo del veicoli con tutte le implicazioni che porta soprattutto in caso di errore del sistema di guida automatizzata. Attualmente i DAS principali, come ad esempio ABS, ESP etc. intervengono nel comportamento del veicolo senza permettere al guidatore di sovrascrivere i comandi, poiché sono caratterizzati da un tipo di percezione molto limitata ma molto precisa, da un sistema decisionale ben circostanziato e ben descritto e sono applicabili solamente in scenari ben definiti; inoltre l'automazione introdotta dai DAS è molto limitata temporalmente e dura solamente alcuni istanti. Gli ADAS al contrario includono una percezione e una attuazione molto complessa, ma i limiti in termini di costo non permettono di avere sistemi di percezione ridondanti e il dominio di applicazione è, come per i DAS, molto limitato e ben circostanziato.

In realtà fra i DAS, gli ADAS e la guida automatica vi sono altri livelli di possibile

automazione e questi sono analizzati nel prossimo paragrafo.

1.2 Analisi dei possibili livelli di automazione

Come introdotto nella sezione precedente, le due tematiche di ricerca principale nel campo dell'ITS possono essere in realtà incluse in una più ampia analisi dei possibili livelli di automazione di un veicolo intelligente. In particolare, 5 livelli possono

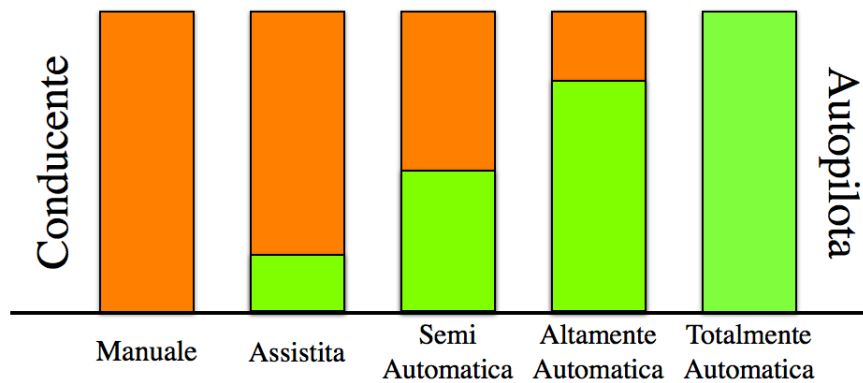


Figura 1.1: Tassonomia nell'automazione di un veicolo.

essere evidenziati[15], come mostrato in figura 1.1:

- Guida Manuale;
- Guida Assistita;
- Guida Semiautomatizzata;
- Guida Altamente Automatizzata;
- Guida Totalmente Automatizzata.

I livelli elencati sono in ordine di crescente automazione, cioè al primo livello si ha il minore intervento dell'automazione mentre nell'ultimo livello si ha il maggiore intervento, come evidenziato anche in figura 1.2 dove la parte arancione di ogni istogramma rappresenta quanta parte del compito di guida spetta al guidatore mentre la parte in verde rappresenta quanta parte spetta all'automazione.

Al primo livello di questa scala si ha la guida totalmente manuale, che non è altro

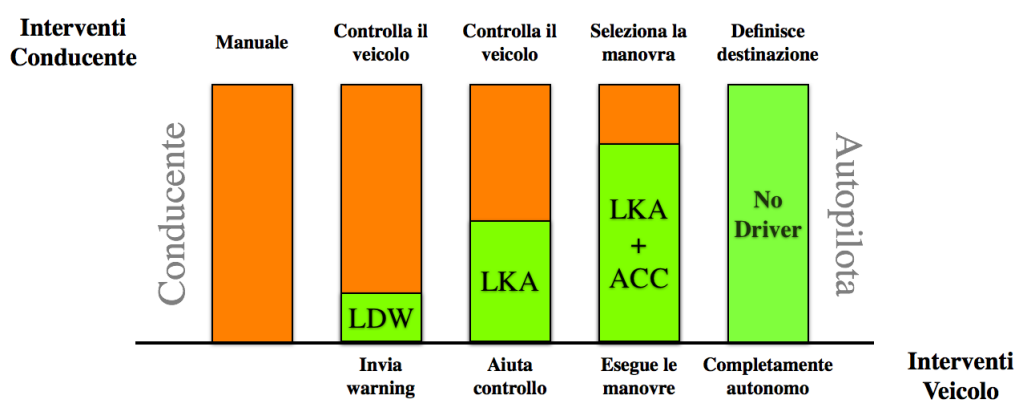


Figura 1.2: Possibili livelli di automazione in un veicolo intelligente.

che il modo in cui si guida ogni veicolo dei giorni nostri, cioè agendo sui classici comandi analogici di sterzo, accelerazione e freno. Nel secondo livello, detto Guida Assistita, il veicolo è in grado di inviare avvertimenti al guidatore, senza però intervenire direttamente nel controllo del veicolo. È questo il caso di un sistema di *Lane Departure Warning* (LDW), cioè di un sistema che avverte il conducente in caso di attraversamento della linea di separazione fra una carreggiata e l'altra; questo sistema infatti avvisa l'utente dell'attraversamento in corso in modo da permettere una adeguata reazione se l'attraversamento non è voluto, ma non interviene direttamente nella guida del veicolo.

La Guida Semi-Automatica è caratterizzata da un limitato intervento da parte dell'automazione nella guida del veicolo: a questo livello infatti non vi sono sempli-

cemente degli *warning* da parte del sistema di guida ma anche un intervento limitato nel tempo. Gli interventi sono spesso motivati da ragioni di sicurezza o di comodità per il guidatore. Gli esempi di applicazioni a questo livello sono:

- Lane Keeping Assistant: questa applicazione aiuta il guidatore a mantenere il veicolo all'interno della corsia attuale, attuando delle leggere correzioni nel caso in cui il veicolo stia per oltrepassare la linea di demarcazione della corsia stessa; allo stesso tempo il guidatore può decidere, agendo sullo sterzo, di sovrascrivere le azioni intraprese dal sistema e cambiare volontariamente corsia.
- Advanced Cruise Control: questo ADAS ha il compito di mantenere il veicolo ad una certa velocità, rispettando allo stesso tempo la distanza di sicurezza dal veicolo che precede.

Il livello successivo, detto Guida Altamente Automatizzata, è caratterizzata da un livello ancora crescente di automazione. In particolare, il guidatore in questo caso seleziona il tipo di automazione desiderato, l'automazione sarà poi responsabile della guida del veicolo anche per lunghi tratti. Gli esempi di questo tipo di guida possono essere molti, ma attualmente nessuno di questi è stato implementato su veicoli commerciali. Un esempio molto interessante è costituito dall'unione dei sistemi di LKS e di ACC: questo nuovo ADAS sarebbe in grado di mantenere il veicolo entro la corsia ad una certa velocità, rispettando allo stesso tempo la distanza di sicurezza rispetto al veicolo che precede. Una applicazione di questo tipo è di grande utilità in situazioni come quella di guida autostradale in cui l'automazione potrebbe farsi carico della guida del veicolo per tratte e tempi anche molto lunghi, alleggerendo di conseguenza il carico di lavoro sul conducente del veicolo.

L'ultimo livello nella scala di automazione è detto Guida Totalmente Automatica; a questo livello il guidatore si deve solo occupare di definire la destinazione del proprio viaggio, poiché il controllo del veicolo è poi totalmente in carico al sistema di guida automatica che ha il compito di pianificare ed eseguire la guida dal punto di partenza alla destinazione scelta dal guidatore.

Come appena visto, al crescere del livello di automazione cambia il tipo di interazione fra il sistema di guida e il conducente: se a bassi livelli di automazione il veicolo intelligente è in grado di proporre solamente suggerimenti o correzioni, ad alti livelli di automazione l'intelligenza artificiale può proporre una serie di soluzioni complete, anche su diversi livelli di complessità, al guidatore; questo porta a capire che al crescere del livello di interazione fra veicolo e guidatore possono sorgere nuovi e diversi conflitti su diversi livelli; il prossimo paragrafo si occuperà dell'analisi di questi conflitti e di una loro possibile soluzione.

1.3 Gestione dei conflitti

Nel primo e nel secondo livello di automazione non possono sorgere conflitti fra il conducente ed il veicolo poiché quest'ultimo non interviene direttamente nel controllo e di conseguenza l'unica interazione possibile è nella gestione della modalità di avviso che il veicolo può dare al guidatore. Dal livello successivo, per contro, l'interazione sale al livello di controllo e di conseguenza possono nascere conflitti fra l'intelligenza artificiale e il guidatore; i conflitti possono essere dovuti ad una diversa intenzione sull'azione da compiere, oppure ad una diversa interpretazione della percezione del mondo esterno od infine ad un errore in una delle due parti.

I conflitti possono avvenire a diversi livelli, come visibile in figura 1.3, in particolare al salire del livello di automazione sale anche il livello della possibile nascita del conflitto; i possibili livelli di conflitto sono:

- Livello di controllo: livello di controllo del veicolo, si intende di questo livello tutto ciò che influisce direttamente sui comandi di accelerazione e sterzo del veicolo stesso.
- Livello di manovra: questo livello di astrazione comprende le manovre che un veicolo può intraprendere, come ad esempio sorpassare un veicolo o mantenere la corsia, oppure cambiare strada.

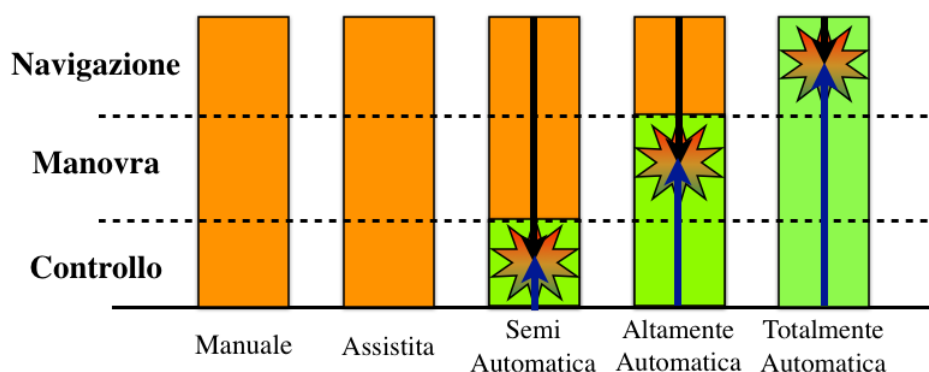


Figura 1.3: Possibili conflitti nei diversi livelli di automazione.

- **Livello di Navigazione:** è il livello di astrazione più alto e comprende principalmente le decisioni su come andare da un punto ad un altro, quali strade intraprendere, senza riguardare direttamente come le strade vengono percorse.

Il primo livello in cui si possono avere conflitti è quello di Guida Semiautomatica. A questo livello il veicolo attiva l'automazione per brevi periodi e l'automazione risiede completamente nel livello di controllo, per cui i conflitti possono avvenire a quel livello; in particolare il veicolo interviene quando rileva un comportamento non sicuro da parte del guidatore oppure quando sopraggiungo situazioni pericolose. Un esempio di conflitto può avvenire durante il sopracitato LKA nel caso in cui il veicolo tenti di assumere una traiettoria errata a causa di una percezione sbagliata delle linee di demarcazione. I conflitti a questo livello possono essere risolti dando al guidatore la possibilità di sovrascrivere i comandi a livello di controllo, ad esempio lasciando che possa riprendere il controllo del veicolo utilizzando i normali comandi di accelerazione e sterzo. Nella Guida Altamente Automatizzata l'interazione fra guidatore e veicolo è diversa da quella appena descritta: guardando la figura 1.4 si considera ad esempio una situazione in cui il guidatore sta guidando manualmente il veicolo; ad un certo punto il sistema di guida automatica individua le condizioni (presenza

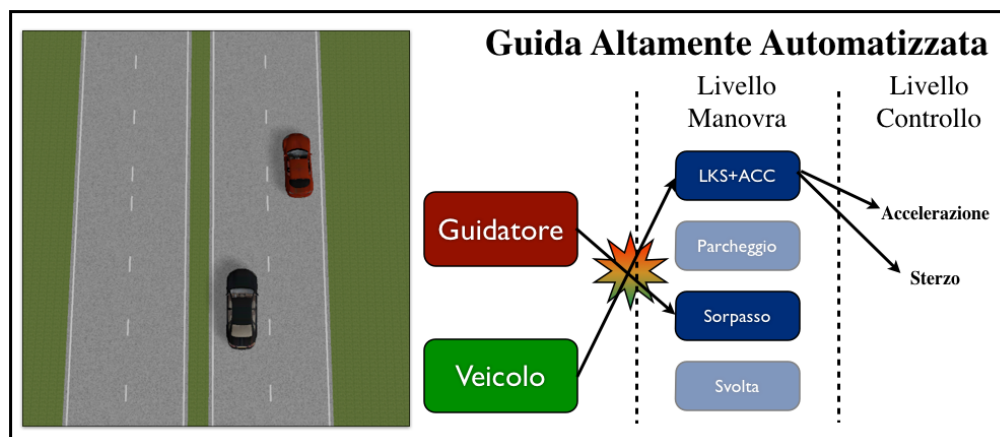


Figura 1.4: Conflitto durante la guida altamente automatizzata.

di linee di demarcazione, rilevamento corretto degli altri veicoli, etc) per la guida altamente automatizzata e propone di conseguenza una serie di manovre possibili, nell'esempio guida in autostrada e sorpasso del veicolo che precede. Di conseguenza il guidatore può decidere di ingaggiare l'autopilota e far sì che il veicolo intelligente continui la guida al suo posto. Il guidatore può decidere in ogni istante di cambiare la manovra corrente selezionando ad esempio la manovra di sorpasso quando vede che il veicolo che precede è molto più lento; d'altro canto il veicolo può giudicare la manovra come non sicura se, come accade in figura 1.4, la percezione rileva un veicolo in arrivo nella corsia sinistra, di conseguenza il veicolo automatico si rifiuterà di compiere la manovra ed nascerà un conflitto fra guidatore e veicolo. Il conflitto in questione, come evidenziato in figura 1.4, risiede al livello di manovra perché nasce da una diversa volontà sul come affrontare un determinato tratto di percorso. A questo punto una qualche forma di risoluzione del conflitto è necessaria, perché non è possibile capire se il conflitto dipende da un guidatore imprudente o se è causato da un errore nel sistema di guida automatizzata e non ci sono abbastanza informazioni per distinguere tra questi due casi. Il conflitto può essere risolto agendo contemporaneamente su diversi livelli: a livello di manovra l'intelligenza artificiale dovrà cambiare

dinamicamente le manovre disponibili in modo da rendere attivabili solo le manovre che giudica, in quell'istante, sicure; d'altro canto il guidatore deve poter mantenere la possibilità di sovrascrivere completamente i comandi a livello di controllo, in modo da poter risolvere tutti i problemi causati da un errore nella percezione del veicolo intelligente.

Nell'ultimo livello di automazione il guidatore si occupa solamente di definire la destinazione del viaggio mentre il veicolo si occupa di tutto il resto, a partire dalla definizione del percorso. Essendo i livelli di manovra e controllo completamente

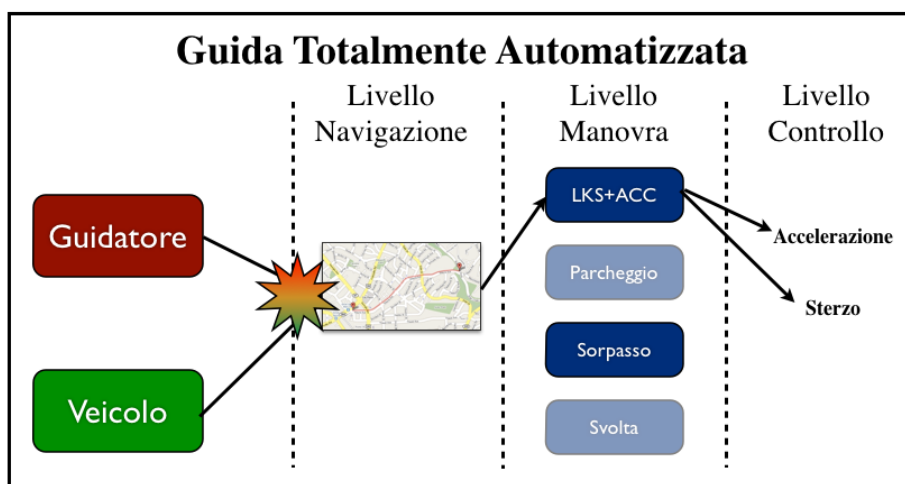


Figura 1.5: Conflitto durante la guida totalmente automatizzata.

gestiti dal veicolo i possibili conflitti nascono a livello di navigazione: è possibile infatti che una volta determinato il percorso il guidatore decida di cambiarlo, sia per un avvenuto cambio di destinazione sia perché il guidatore decide di intraprendere una strada diversa, come per esempio accade in figura 1.5 in cui viene evidenziato come i conflitti avvengano a livello di navigazione. Nasce a questo livello la necessità di imporre un arbitrio: la soluzione risiede nel risolvere il problema su un duplice livello infatti i problemi a livello di navigazione dovranno essere risolti attraverso un apposita interfaccia che permetta all'utente di poter scegliere la strada che ritiene più

opportuna mentre tutti gli errori dovuti ad un problema di percezione dell'ambiente o ad un calcolo errato della traiettoria del veicolo devono essere risolvibili dando all'utente la possibilità di sovrascrivere i comandi a livello di controllo in modo che esso possa prendere il pieno controllo del veicolo in caso di necessità.

Concludendo, i possibili conflitti fra guidatore e veicolo sono risolti in modo diverso dipendentemente dal livello di automazione in cui sorgono, in particolare i conflitti possono essere risolti a livello di controllo, manovra o navigazione, utilizzando una Human-Machine Interface (HMI) diversa per ogni livello. Di seguito viene data una descrizione generica dei meccanismi di HMI e una proposta di soluzione per i casi analizzati.

1.3.1 Descrizione delle Interfacce uomo macchina

I principi delle interfacce uomo-macchina sono applicabili a molto aspetti dell'attività di guida e devono essere considerati nel design di tutti i sistemi a bordo veicolo, in particolare degli ADAS. Alcune ricerche, come ad esempio in [16], mostrano un impatto negativo dei sistemi di assistenza alla guida che risultano distrarre particolarmente le persone più anziane.

Le HMI per veicoli intelligenti devono includere tre modi di scambio informazioni fra il guidatore ed il veicolo:

- Dare un feedback al guidatore.
- Ricevere un input dal guidatore.
- Eseguire azioni in autonomia.

Feedback al guidatore

Dare un feedback al guidatore significa comunicare con esso utilizzando i canali sensoriali, in sintesi significa mandare messaggi che possono essere:

- Visuali
- Auditivi

- Tattili
- Olfattivi

Ogni tipo di messaggio richiede un differente tempo di percezione; il tempo di percezione è influenzato anche dalla complessità del messaggio e dal numero di compiti che in quel momento il destinatario sta svolgendo. La vista è un senso che viene attivato da tutti i tipi di messaggio, infatti uno stimolo ricevuto su uno qualsiasi dei quattro sensi elencati sopra causa una ricerca visuale aggiuntiva per capire la sorgente dello stimolo. Ad esempio il comparire di un segnale visivo o acustico all'interno di un autoveicolo causa la ricerca, da parte del guidatore, della sorgente di quel segnale e questo viene fatto principalmente utilizzando il senso della vista e causa l'aumento del carico di lavoro mentale che il conducente sta eseguendo. In generale tutti gli stimoli, generati su tutti i canali sensoriali, contribuiscono al carico di lavoro del conducente; ogni guidatore ha inoltre la capacità di distogliere temporaneamente l'attenzione dalla strada senza per questo distrarsi, detta capacità visuale, purché il cambio di focus dell'attenzione sia molto limitato nel tempo. Si ha distrazione visuale quando lo stimolo visivo eccede la capacità visuale, quando cioè si sta troppo a lungo concentrati su un compito diverso dalla guida o fissando un oggetto che non sia la strada.

Per quanto riguarda i messaggi acustici essi sono meno impegnativi dal punto di vista del carico mentale poiché i guidatori abitualmente compiono attività come ascoltare la radio mentre guidano, ma è facile capire che messaggi acustici particolarmente complessi potrebbe causare una distrazione dell'attenzione del guidatore. Uno dei sovraccarichi più difficili da misurare è quello cognitivo, che accade principalmente quando il guidatore è distratto nel pensare a cose diverse rispetto all'attività di guida. Un ulteriore sovraccarico considerabile è quello manuale, che accade quando il guidatore occupa le mani in azioni secondarie, come bere un caffè o sintonizzare la radio; anche per questo sovraccarico vale la considerazione fatta per i sovraccarichi visuali in quanto l'utente ha una certa capacità di effettuare azioni manuali secondarie senza distrarsi dall'attività di guida, ma se queste azioni superano la capacità massima dell'utente si ha la distrazione dell'attenzione del guidatore.

I feedback aptici fanno parte del design standard di un veicolo che ne permette il controllo: il feedback di forza sulla sterzo e la sua risposta agli input dell'utente rappresentano una delle caratteristiche nel design di un veicolo. Altri feedback aptici, anche se in senso lato, sono la risposta del pedale del gas e del freno, oltre alle vibrazioni introdotte dal veicolo che in veicoli di lusso sono sicuramente minori rispetto ad un veicolo economico e che concorrono alla comodità della guida e di conseguenza alla stanchezza e al carico di lavoro a cui è soggetto il guidatore. I sistemi di feedback aptici possono essere dei validi metodi per dare avvertimenti al conducente, soli od in combinazione con sistemi visivi o acustici. Un esempio è l'utilizzo della vibrazione sul volante per avvertire il guidatore in caso di attraversamento della corsia; questa emulazione delle strisce vibranti autostradali ha dimostrato di essere molto efficace e di causare un minimo aumento del carico di lavoro mentale.

I feedback olfattivi non sono attualmente utilizzati all'interno delle interfacce uomo macchina, anche se degli studi empirici hanno dimostrato che la stimolazione olfattiva può aumentare le performance dei feedback tattili[17].

Una ulteriore considerazione nel design delle HMI per sistemi veicoli intelligenti deve essere la distribuzione, anche temporale, degli azioni visive, acustiche e manuali che il veicolo si aspetta dal guidatore, poiché è stato dimostrato che il guidatore può svolgere diversi compiti visivi nello stesso momento mentre i compiti manuali sono sequenziali. Ne consegue che gli avvertimenti integranti diverse modalità sensoriali possono essere utili se non opprimo il conducente; infatti l'aggiunta di feedback aptici agli avvisi audio/visuali può aumentare l'efficacia di questi ultimi senza influenzare eccessivamente il carico mentale del conducente. In generale, gli avvisi integrati devono tener conto dei bisogni manuali, percettivi e cognitivi del guidatore, e, pur non esistendo regole assolute nel design degli avvisi per gli ADAS, tener conto di quanto appena detto può portare alla costruzione di HMI più performanti dal punto di vista del tempo di reazione del conducente.

Input dal guidatore

Una HMI per veicoli intelligenti deve essere in grado di ricevere input dall'utente e di eseguire i comandi che vi sono contenuti. Le HMI di basso livello normalmente sono

composte da tutti quei comandi a bordo veicolo che permettono di controllare analogicamente il veicolo, ad esempio lo sterzo, l'acceleratore ed il freno. I comandi dati attraverso queste interfacce possono essere aumentati dal veicolo, come ad esempio accade nelle applicazioni di *brake assist* in cui il veicolo riceve un input iniziale di frenata e, se percepisce che l'input è inadeguato, aggiunge forza franante sul freno in modo da avere una frenata sicura ed efficace.

Per quando riguarda le HMI per alti livelli di automazione è possibile utilizzare input che siano sia manuali che vocali; molti ADAS odierni sono infatti attivabili vocalmente con sistemi che prevedono anche l'addestramento dell'auto ai comandi del conducente. L'utilizzo combinato di comandi manuali e vocali deve servire a ridurre il carico di lavoro richiesto al conducente per l'utilizzo di tutti i sistemi di guida intelligenti disponibili. Infine è consigliabile attuare dei meccanismi atti a riunire e ordinare l'attivazione degli ADAS in modo da renderli facilmente gestibili a tutti i tipi di guidatori.

Azioni in autonomia

Le interfacce uomo macchina hanno anche il compito di eseguire azioni in autonomia, cioè eseguire determinate azioni senza richiedere alcun intervento specifico da parte dell'utente. Ad esempio questo accade in caso di guida completamente autonoma quando a seguito di un input dell'utente il veicolo si gestisce in completa autonomia senza richiedere altri input da parte dell'utente, oppure negli ADAS quando i comandi dell'utente sono aumentati dalle capacità di automazione del veicolo, come visto anche nella sezione precedente. Da tenere in considerazione il fatto che il sistema di guida automatizzata potrebbe aver bisogno di un feedback da parte dell'utente, ed in generale idealmente avrebbe bisogno di sapere in ogni momento qual'è lo stato di attenzione del guidatore. Lo stato di attenzione si riferisce a quanta attenzione il guidatore sta impiegando nel compito primario di guida o di sovrintendenza alla guida. Durante una sessione di guida normale lo stato di attenzione è manifestato principalmente dallo sguardo del guidatore che deve essere fisso sulla strada; una deviazione da questa postura dello sguardo può indicare la distrazione del guidatore, così come l'apertura/chiusura dei suoi occhi o gli sbadigli. Molti fattori come la fatica, la son-

nolenza o altri disturbi esterni possono causare una degradazione dell'attenzione del conducente. In sessioni di guida altamente automatizzata è normale che il conducente ponga una minore attenzione nel compito di guida, ma il veicolo deve essere a conoscenza di quanto il guidatore è attento per attuare le opportune politiche di richiamo in caso di necessità.

1.3.2 HMI per diversi livelli di automazione

In una architettura che supporti tutti i livelli automazione si avrà bisogno quindi di 3 diverse HMI:

- Control Level HMI: tipicamente i comandi di sterzo e accelerazione/freno, in un veicolo intelligente possono essere modificati per rilevare l'intenzione del guidatore di intervenire a livello di controllo, tramite appositi pulsanti posti sul volante o tramite un sensore in grado di valutare la forza impressa sui comandi da parte del guidatore. Allo stesso tempo possono essere mandate tramite HMI feedback e avvisi integrati che uniscono i messaggi acustici, visuali e tattili al fine di facilitare la comunicazione fra veicolo e guidatore.
- Maneuver Level HMI: touch screen tramite il quale è possibile scegliere tra le manovre proposte dall'intelligenza artificiale, oppure una serie di comandi posti sul volante che permettano di scegliere la manovra desiderata. A questo livello è possibile integrare sistemi di input quali il comando vocale ed utilizzare display di tipo *head-up* per facilitare la comunicazione e diminuire il carico cognitivo sul guidatore.
- Navigation Level HMI: touch screen tramite il quale sia possibile scegliere il percorso che si desidera oppure la possibilità di cambiare virtualmente strada utilizzando lo sterzo come indicatore di direzione, anche a questo livello è possibile inserire metodi di input vocale e schermi di tipo *head-up* per facilitare la comunicazione e diminuire il carico cognitivo del guidatore.

1.4 Guida a livello di manovra

Introduciamo adesso l'idea base di questa tesi di dottorato: visti i problemi ed i conflitti di ogni livello di automazione si è deciso di sviluppare e progettare una nuova architettura in grado di supportare i livelli di automazione visti in precedenza; inoltre si vuole sviluppare un nuovo paradigma di guida a livello altamente automatizzato, che chiameremo *Digital Driving*, che consenta di guidare un autoveicolo con comandi discreti ad alto livello di astrazione, detti manovre, senza interagire direttamente con i normali comandi di guida ma lasciando il pieno controllo al veicolo intelligente per lunghi tratti di percorso.

I vantaggi di questo tipo di guida rispetto agli altri livelli di automazione sono i seguenti:

- Conducente nel ciclo percezione-azione;
- Esclusione del conducente dalla guida diretta del veicolo;
- Raggruppamento di tutti gli ADAS entro un'unica HMI;
- Fattibilità legale del progetto;

Dal punto di vista della guida totalmente automatizzata vista nell'analisi dei diversi livelli di automazione l'interazione fra il veicolo ed il conducente avviene al livello di navigazione, come già evidenziato nel paragrafo 1.2; in questo caso il guidatore assegna uno o più punti di destinazione e l'automazione del veicolo ha il compito di svolgere in maniera sicura ogni manovra che porti dal punto di partenza verso tutti i punti di destinazione. Questo significa che il sistema di guida automatica deve prendere ogni decisione senza poter far affidamento in alcun modo sul conducente, il quale, in accordo con il concetto di guida totalmente automatizzata, non rientra nel ciclo di azione-percezione. Di conseguenza, si ha la necessità di avere sistemi di percezione e sistemi cognitivi che siano molto performanti, soprattutto in termini di comprensione dell'ambiente che circonda il veicolo.

Come è stato mostrato nell'introduzione di questo capitolo la percezione del mondo esterno e l'interpretazione delle varie situazioni di guida è sicuramente il compito

più complesso di un sistema di guida intelligente. Potenzialmente esistono due tipi di errore nell'interpretazione delle situazioni di guida: falsi positivi (detti anche α errors) e falsi negativi (detti anche β errors). In caso di falso positivo l'automazione rileva un ostacolo quando in realtà non ve ne sono, mentre per i falsi negativi non viene rilevato un oggetto dove in realtà ne è presente uno. La sfida nel design delle architetture di sistema consiste nel fatto che un sistema percettivo che riduce uno dei due tipi di errore automaticamente aumenta l'altro. Le architetture di sistema per veicoli automatici normalmente si basano su complesse politiche di sistema, come descritto in [18] e su decisioni strategiche conservative tendenti ad ammettere più falsi positivi che falsi negativi; questo perché i falsi positivi sono intrinsecamente meno pericolosi dal punto di vista della sicurezza per eventuali passeggeri. In conclusione, le strategie di recupero automatico da situazioni di errore sono molto dispendiose dal punto di vista della complessità computazionale poiché devono valutare molte diverse alternative con un potenziale di rischio che incrementa di volta in volta, com'è mostrato in [19]. Questo processo decisionale può essere significativamente abbreviato inserendo un guidatore nel ciclo di percezione azione, infatti le intrinseche abilità di ogni conducente di decidere quando un rilevamento positivo è vero o falso è superiore a quello dell'automazione. Inoltre, un conducente è in grado di riconoscere e analizzare velocemente ogni situazione di guida, anche complessa e in caso di necessità è in grado di derivarne le alternative. Quindi l'integrazione del conducente all'interno del ciclo di percezione-azione è uno dei grandi vantaggi di una guida a livello di manovra come quella progettata in questa tesi.

In contrasto a questo si consideri che, prendendo ad esempio gli ADAS attuali, la loro tecnologia potrebbe essere in grado di guidare senza l'intervento umano, almeno in alcune situazioni ben circostanziate come in situazioni guida autostradale. In questo caso infatti un veicolo equipaggiato con un sistema di *Adaptive Cruise Control* (ACC) e un sistema di *Lane Keeping Assistant* (LKA) è in grado di effettuare sia il controllo longitudinale che il controllo laterale del veicolo, ed è quindi in grado, già adesso, di guidare in un ambiente limitato come quello autostradale. Anche negli ADAS, come del resto già evidenziato per la guida completamente automatica, è possibile incorrere in comportamenti errati e di conseguenza è necessaria la supervisione

	<i>DAS base</i>	<i>DAS avanzati</i>
Percezione	Minima e molto focalizzata	Complessa
Automazione	Di breve durata e iniziata automaticamente	Lunga durata e attivata dall'utente
Supervisione	Non Necessaria	Necessaria

Tabella 1.1: Differenze fra DAS e ADAS.

da parte del guidatore: se il sistema è consapevole dei proprio errori potrà informare il guidatore e rilasciare, con le dovute precauzioni, il controllo al guidatore stesso; viceversa in caso contrario sarà il guidatore a doversi accorgersi dell'errore e a riprendere il controllo del veicolo utilizzando i normali controlli analogici. Per questo motivo l'attivazione e la supervisione dei DAS base e avanzati differiscono fra loro, come del resto evidenziato dalla tabella 1.1. In particolare, per alcuni DAS l'automazione è molto breve e di conseguenza non richiedono supervisione in quanto la percezione è minima e molto focalizzata. D'altro canto gli ADAS devono affrontare situazioni anche molto complesse in una automazione che può essere attivata anche per lunghi tratti; in questo caso il guidatore deve supervisionare il sistema.

La guida analogica propria della auto dei giorni nostri ha due problemi principali:

- Necessità di addestramento alla guida.
- Vulnerabilità del fattore umano.

In primo luogo l'attività di guida non è naturale, di conseguenza ogni guidatore deve essere addestrato alla guida di un autoveicolo. Come ogni attività soggetta ad addestramento le capacità di ogni soggetto influiscono sulle prestazioni finali, di conseguenza vi saranno alcune categorie di guidatori meno abili di altre e più soggette ad errori ed a causare incidenti stradali nei casi in cui il compito della guida diviene più complesso, come in caso di manto stradale accidentato o come in caso di situazione meteorologica avversa. Inoltre ogni guidatore è vulnerabile a fattori personali e

ambientali: la stanchezza, come la sonnolenza o gli stato emotivi come la preoccupazione o l'alterazione dovuta ad alcool o droghe possono influenzare pesantemente l'attività di guida e comprometterne la sicurezza. Si ricorda, come già accennato nell'Introduzione a questa tesi di dottorato, che circa il 92% degli incidenti stradali sono dovuti al fattore umano; di conseguenza appare evidente come l'esclusione del fattore umano dalla guida diretta del veicolo rappresenti un notevole punto di forza della guida a livello di manovra.

Il crescente numero di ADAS sui veicoli della prossima generazione pone un ulteriore problema: questi sistemi sono eterogenei ed indipendenti l'uno dall'altro, di conseguenza possono causare un sovraccarico nelle attività del guidatore che deve supervisionare tutti gli ADAS con un conseguente aumento della difficoltà dell'attività di guida da parte del guidatore. Inoltre, come evidenziato nella sezione 1.3.2, interfacce uomo macchina eterogenee e separate portano un grande sovraccarico nell'attività cognitiva poiché non tengono conto delle necessità fisiche, cognitive e manuali che ogni guidatore ha. Di conseguenza il raggruppamento di tutti gli ADAS all'interno di un unico paradigma di guida a livello di manovra porta come grande vantaggio la semplificazione dell'interfacciamento fra uomo e macchina, poiché tutte le interfacce sono riunite in una unica HMI, con conseguente scarico delle attività cognitive del guidatore.

Come ultimo fattore si consideri che secondo la convenzione sul traffico stradale di Vienna del 1968 [20], che è la convenzione europea attualmente in vigore su cui sono basate le legislazioni delle nazioni europee, ogni veicolo necessita di un guidatore che sia responsabile del veicolo in caso di incidente, anche durante le sessioni autonome; di conseguenza per rendere legalmente fattibile una guida altamente automatizzata è necessario considerare un guidatore come supervisore e responsabile dell'automobile. Gli aspetti legali relativi alla guida altamente automatizzata sono analizzati più nel dettaglio nell'appendice A.

1.4.1 Conduct-by-Wire

Vengono ora analizzate due implementazioni, appartenenti allo stato dell'arte, di guida altamente automatizzata. La prima implementazione è detta *Conduct-by-Wire*

(CbW) ed è stata proposta da un gruppo di ricerca appartenente alla TU Darmstadt [21]. L'idea alla base del CbW è di spostare la guida di un veicolo dal livello di stabilizzazione al livello di manovra includendo una interazione fra guidatore e veicolo basata sulla manovra. Di conseguenza in questo modo la convenzionale e continua interazione fra guidatore e veicolo viene sostituita da una comunicazione basta sugli eventi a livello di manovra.

In figura 1.6 è raffigurata l'architettura di sistema ad alto livello di CbW. Il guidatore

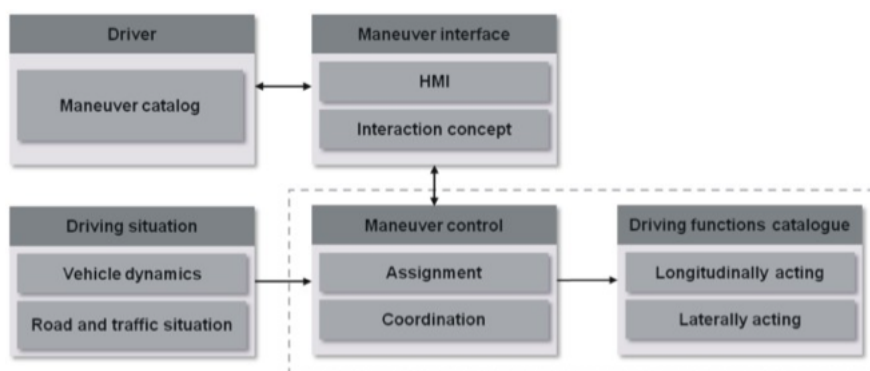


Figura 1.6: Architettura di sistema di Conduct-by-Wire.

assegna i comandi di manovra attraverso il componente chiamato "Maneuver Interface", la quale permette la parametrizzazione della manovra scelta e l'interazione a livello di stabilizzazione se necessario. Questa interfaccia rappresenta l'HMI principale del sistema CbW. Il componente successivo nell'architettura di figura 1.6 è il controllo di manovra il quale, dipendentemente dalla situazione corrente di guida, assegna l'ordine di manovra dato dal conducente ad esattamente una coppia di funzioni di guida longitudinale e laterale, come descritto in [22].

Catalogo delle manovre

Una delle componenti fondamentali del sistema CbW è il catalogo delle manovre. Nelle intenzioni degli sviluppatori questo catalogo è costruito dal punto di vista del

conducente; inoltre Conduct-by-Wire distingue tra manovre implicite ed esplicite. Le manovre dette esplicite sono unità operative chiuse, che sono assegnate dal guidatore al sistema (ad esempio cambia corsia destra/sinistra); le manovre implicite sono unità operative di durata non limitata (ad esempio: segui la corsia) che vengono invocate automaticamente dopo una manovre esplicita. Ad esempio dopo un "cambia corsia sinistra" viene automaticamente invocato un "segui corsia". Ne consegue che il conducente, non dovendo assegnare direttamente le manovre implicite, può utilizzare un catalogo di manovre ridotto in cui le manovre presenti sono, almeno nell'intenzione degli sviluppatori, delle traduzioni dirette delle più comuni intenzioni. Ad esempio se il conducente ha l'intenzione di girare a sinistra deve selezionare l'apposita manovra. In questo modo il guidatore può pensare in modo "naturale", cioè in termini di obiettivi senza necessariamente pensare a come raggiungere gli obiettivi stessi. Un

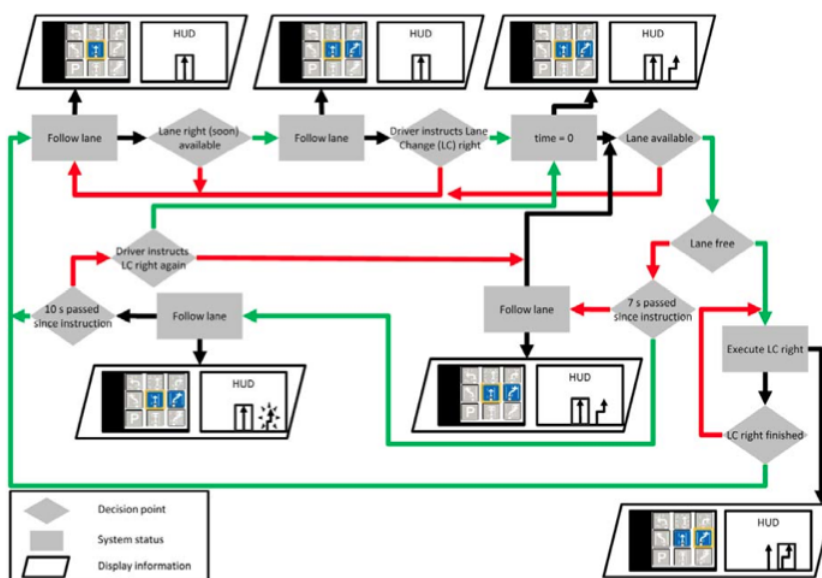


Figura 1.7: Esempio di manovra in CbW: cambio corsia destro.

esempio di come una manovra è implementata è rappresentata in figura 1.7, trattando

la manovra di "Cambio corsia destro". Per ogni manovra di cambio corsia la condizione base è quella di un veicolo in movimento; successivamente viene valutata in varie fasi la condizioni di corsia destra libera. Il diagramma in figura 1.7 descrive anche tutti gli aggiornamenti della HMI che avvengono ad ogni cambio di stato interno della manovra.

Conduct-by-Wire HMI

In Conduct-by-Wire è stata posta grande attenzione nello studio delle HMI; l'interfaccia è stata progettata come combinazione di un volante ed un touch screen, come mostrato in figura 1.8. Il volante, in combinazione con la pedaliera, è destinato ad essere l'elemento di controllo per ogni compito di stabilizzazione; l'assegnazione delle manovre è effettuata utilizzando il display touch screen al centro del volante stesso. Il volante, oltre alle classiche funzioni di controllo, è dotato di un feedback aptico

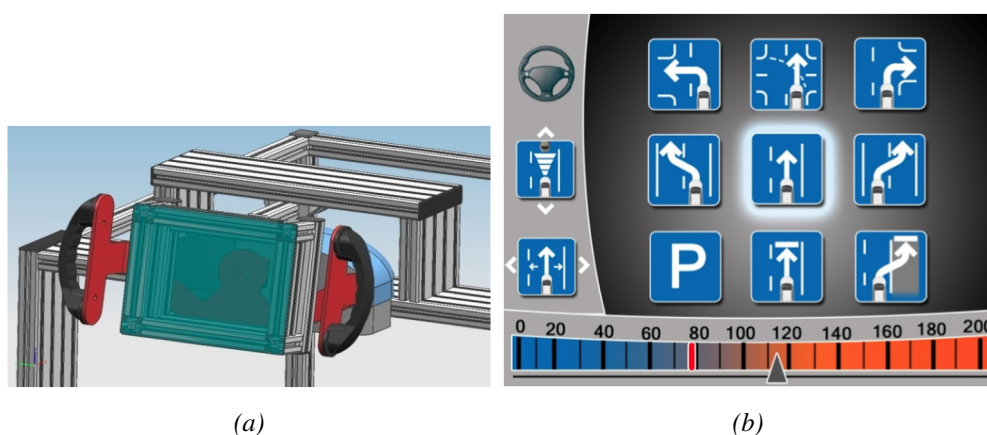


Figura 1.8: (a) Rappresentazione CAD della HMI scelta in CbW (b) User interface in CbW.

limitato allo stato dell'asfalto ed è in grado di modificare la propria risposta in relazione alla velocità di crociera del veicolo. Il display touch ha la funzione di mostrare, in maniera adattativa, le manovre fattibili. La lista delle manovre fattibili è ricavata valutando le condizioni atmosferiche e stradali. Il design della *Graphical User Inter-*

face è statico in modo da aumentare la raggiungibilità di ogni elemento all'interno della GUI.

La GUI, di cui si può vedere un esempio in figura 1.8(b) è costituita da una griglia 3x3 di icone in cui sono raffigurate tutte le manovre disponibili; nella parte sinistra sono riportati due parametri di configurazione e il meccanismo di avvio del sistema. Nella parte bassa dello schermo è riportato un tachimetro che segnala l'attuale velocità e la velocità richiesta.

1.4.2 H-Metaphor

Nel 2003 Flemish et Al. [23] hanno proposto una modalità di interazione fra uomo e veicolo detta *Horse Metaphor*. L'idea base consiste nel fatto che l'uomo è già in grado di viaggiare su un veicolo intelligente: il cavallo. Durante un viaggio su un cavallo il conducente non deve occuparsi di dare un input continuo all'animale, né deve preoccuparsi di guidare l'animale in modo che eviti gli ostacoli o in modo che non esca di strada.

Il cavallo infatti è un mezzo di trasporto intelligente e non ha bisogno che il guidatore pensi alla sicurezza di se stesso e del mezzo, poiché è in grado in totale autonomia di percorrere dei percorsi senza danneggiare se stesso né altri. Durante la guida di un cavallo inoltre il guidatore è sempre a conoscenza dello stato dell'animale e del sopraggiungere di eventuali pericoli attraverso il contatto aptico dato dalla montatura del cavallo. Un'altra caratteristica molto interessante della guida di un cavallo sta nel fatto che il cavallo può essere portato in due modalità:

- Redini strette: il conducente tendendo le briglie comunica al cavallo che vuole guidarlo direttamente passo passo. Durante il viaggio con redini strette il cavallo perde in parte la propria indipendenza e viene guidato completamente dal conducente;
- Redini sciolte: allentando le redini il conducente comunica al cavallo che può camminare in autonomia e che non lo guiderà più direttamente; questo fa sì che il cavallo si prenda la responsabilità del normale incedere mentre il conducente

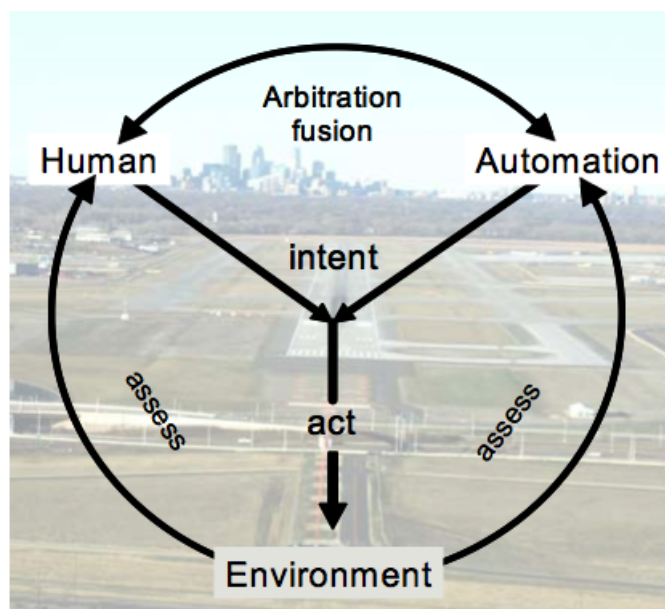


Figura 1.9: Modello di interazione fra due agenti intelligenti.

si occupa solamente di dare una indicazione generale di direzione e velocità alla propria montatura.

Si noti come le modalità a redini strette o larghe sono in realtà i due estremi di quello che può essere considerato più una funzione continua che due stati discreti.

La H-metaphor appena riassunta è stata utilizzata per strutturare una relazione tra un operatore e un veicolo intelligente (un aereo, in particolare) in [24]. L'obiettivo è far cooperare due agenti intelligenti con caratteristiche diverse, come mostrato in figura 1.9. Implementando la H-metaphor la componente umana si occuperà di studiare le strategie, di confermare le decisioni dell'altra componente intelligente e di correggerla in caso di problemi in quelle azioni che vanno oltre le capacità cognitive autonome del veicolo. Il veicolo è aiutato nel suo ruolo dalle capacità del veicolo intelligente, il quale è in grado di eseguire tutte le operazioni di routine, è in grado di effettuare pianificazione e controllo di traiettoria, infine è in grado di controllare e

mantenere la sicurezza operativa.

L'obiettivo di questa interazione è aumentare sicurezza, performance e flessibilità complessive del sistema oltre il livello che sarebbe raggiungibile con la tradizionale automazione e diminuendo sensibilmente i costi di addestramento dell'operatore. In aggiunta con l'introduzione di questa metafora è possibile definire un percorso con capacità di automazione incrementali che possa essere una alternativa ai veicoli completamente autonomi. Il metodo H-metaphor si configura quindi come una gui-

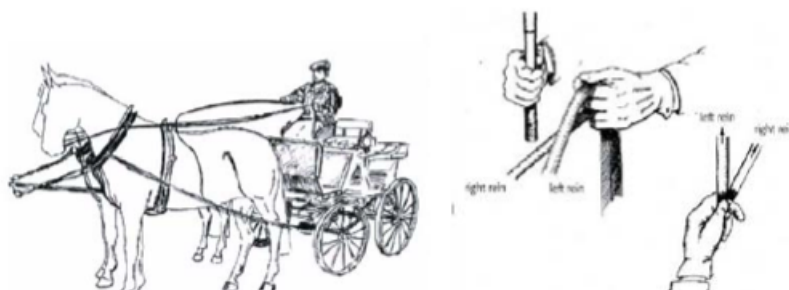


Figura 1.10: Connessione guidatore cavallo nel sistema con carretto.

da nell'interazione fra guidatore e veicolo, considerando il veicolo come un agente autonomo, e come una guida nell'interazione multiveicolo [23]. Il cavallo e il suo guidatore interagiscono tramite una comunicazione multi-modale consentita da un forte e bi-direzionale contatto aptico; il contatto aptico è unico in questo senso poiché consente allo stesso tempo di percepire e di comunicare. In caso di cavallo con carretto il contatto aptico è sostituito dalle redini, le quali formano una comunicazione bidirezionale con il cavallo con lo scopo di ricevere e trasmettere messaggi fra i due agenti intelligenti, come mostrato in figura 1.10. Quest'ultimo caso è particolarmente interessante, in quanto risulta essere molto più simile al caso di connessione conducente-veicolo; il contatto aptico fra conducente e cavallo non è infatti giudicato implementabile su un veicolo intelligente.

In questo tipo di interazione ci deve essere sempre una "connessione morbida, costante ed elastica" tra il conducente e il cavallo, dove "il cavallo cerca il contatto

ed il conduttore glielo fornisce” [25]. Nel caso in cui il cavallo volesse cambiare direzione, per esempio per questioni di sicurezza, può comunicarlo al conducente con un movimento della testa; il conduttore può percepire il cambio di tensione nelle redini e può accettare o rifiutare l’azione rilassando o incrementando la tensione nelle redini stesse. In caso di una mancanza di tensione sulle redini il cavallo può capire che il guidatore è relativamente disinteressato all’azione di guida e modificare il proprio comportamento di conseguenza.

L’automazione convenzionale tende ad essere digitale nel controllo del veicolo, infatti il controllo può essere posseduto dal veicolo o dal guidatore, ma mai da entrambi contemporaneamente. Nell’approccio basato sulla H-metaphor il controllo può essere posseduto dal conducente mentre il cavallo esegue altre operazioni, come ad esempio il controllo dei pericoli circostanti, oppure il controllo può essere in mano al cavallo mentre il conducente esegue altre operazioni, tra cui la supervisione sulle operazioni del cavallo stesso. Nel primo caso si riflette il caso di redini strette, nel secondo caso si ha un’implementazione delle redini sciolte. Come ultima analisi della H-metaphor viene riportata una comparazione delle comunicazione in diverse modalità di guida in tabella 1.2.

Interazione	Veicolo Convenzionale	Veicolo Con- venzionale Avanzato	Veicolo ispirato ad H-metaphor
Direzione	Unidirezionale	Principalmente unidirezionale	Principalmente Bidirezionale
Linguaggio	Analogico/Spaziale	Analogico spaziale con componenti astratte	Analogico/Spaziale
Modalità	Unimodale con componenti aptiche	Visuale con aspetti multi modali	Multi modale con aspetti fortemente aptici
Discreto/ Continuo	Continuo	Discreto	Mix continuo discreto

Importanza della modalità visuale	Alta	Media	Media
Ridondanza nella interazione	Bassa	Media	Alta
Negoziiazione	Nessuna	Poca, volontà dell'automazione nascosta	Presente e trasparente
Loop fisico	Guidatore	Guidatore/automaz. esclusivamente	Guidatore e automazione contemporaneamente

Tabella 1.2: Confronto fra diverse modalità di guida automatizzata.

Capitolo 2

Contributo alla guida a livello di manovra

Chiedersi se un computer possa pensare non è più interessante del chiedersi se un sottomarino possa nuotare.

– Edsger Dijkstra

2.1 Digital Driving

Come visto nel capitolo precedente in letteratura sono presenti diverse soluzioni che cercano di dare una soluzione al problema della guida a livello di manovra. Riassumendo, in [23] viene presentata un'approccio all'interazione uomo-veicolo basato sulla *horse metaphor*; il fulcro di questo approccio consiste nello sviluppo di una HMI aptica multimodale che permetta al guidatore di comunicare specifiche azioni al veicolo e che allo stesso tempo permetta al veicolo di comunicare con il guidatore. Un diversa soluzione è presentata in [26] dalla TU Darmstad ed è basata sul paradigma del Conduct-by-Wire: un insieme finito di elementi, chiamati manovre, viene usato per pilotare il veicolo. Ogni manovra viene tradotta direttamente in azioni dal-

l'automazione senza nessun intervento da parte del guidatore sui tradizionali controlli di guida.

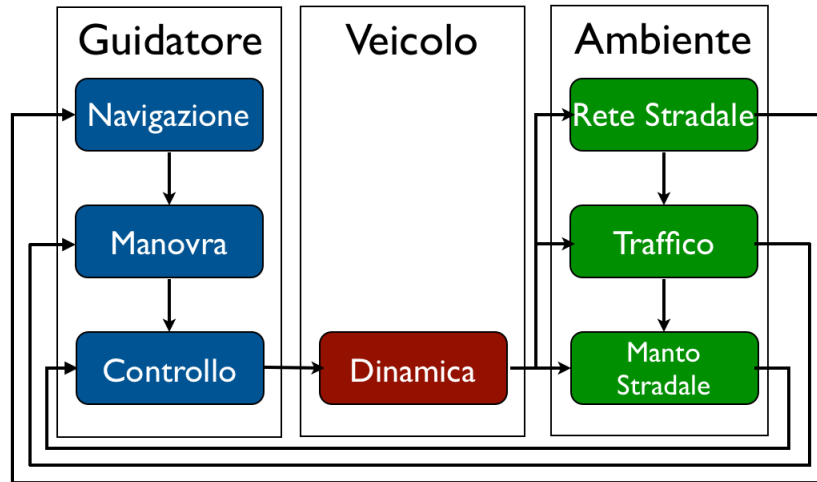


Figura 2.1: Veicolo convenzionale: divisione in layer.

Il contributo di questa tesi di dottorato alla guida a livello di manovra è costituito da un nuovo approccio alla guida altamente automatizzata, chiamato Digital Driving, basato su una continua interazione fra uomo e veicolo. Fanno parte di questo approccio un catalogo di azioni di alto livello, una politica di risoluzione dei conflitti fra i due agenti intelligenti e una adeguata HMI di utilizzo.

L'attività di guida può essere divisa in tre principali livelli gerarchici [27]: navigazione, guida (manovra) e stabilizzazione. Al livello di navigazione il conducente sceglie un percorso adeguato che gli consenta di raggiungere la propria destinazione; questo compito è normalmente svolto all'inizio dell'attività di guida e può essere ripetuto in caso di problemi intercorsi durante il tragitto. A livello di guida il conducente produce tutte le traiettorie necessarie a percorrere le strade pianificate al livello superiore; queste azioni non influiscono sulla pianificazione fatta precedentemente. L'ultimo ed inferiore livello, detto livello di stabilizzazione, riguarda il corretto posizionamento del veicolo nella carreggiata; ad ogni istante la posizione viene rifinita

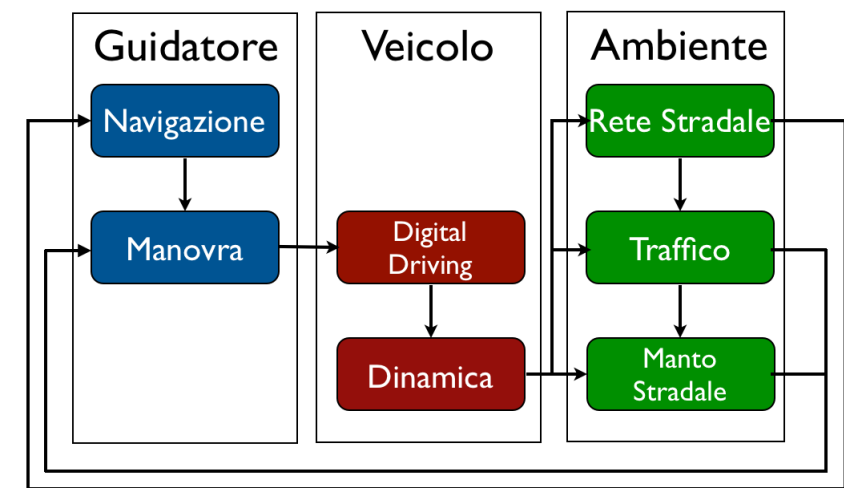


Figura 2.2: Veicolo con Digital Driving: divisione in layer.

dal guidatore utilizzando i tradizionali comandi di accelerazione, freno e sterzo.

Nelle auto tradizionali, come analizzato in figura 2.2, ogni conducente opera su tutti e tre i livelli contemporaneamente: inizialmente viene pianificato il viaggio e le eventuali strade alternative (navigazione), successivamente viene scelta una adeguata traiettoria e velocità per ogni strada da percorrere (guida o manovra); infine queste ultime vengono trasformate in azioni concrete con movimenti dello sterzo o accelerazioni/frenate(stabilizzazione).

L'obiettivo del Digital Driving consiste nel promuovere l'interazione fra conducente e veicolo dal livello di stabilizzazione al livello di manovra, permettendo di andare oltre al controllo diretto del veicolo attraverso le normali interfacce analogiche; oltre a questo si vuole inserire questo paradigma di guida all'interno di una architettura generica che possa supportare molti livelli di automazione e che possa trarre vantaggio dalla cooperazione intra-livello. Grazie ad un set di manovre molto ridotto, che sarà analizzato nel seguito, ogni situazione di guida sarà ben descritta da una sola manovra senza che il conducente debba preoccuparsi di come questa azione verrà implementata. Durante la progettazione e l'implementazione del para-

Il fattore umano è stato tenuto all'interno del loop percezione-azione in modo da implementare una cooperazione fra uomo e macchina, mantenendo al contempo il guidatore responsabile delle attività del veicolo; questo è stato reso possibile in particolare grazie alla creazione di una politica di risoluzione dei conflitti che regola i rapporti fra gli agenti intelligenti.

Il Digital Driving è costituito principalmente da 3 elementi:

- Catalogo delle manovre: catalogo in cui sono definite tutte le possibili azioni implementabili dal veicolo.
- Human-Machine Interface: mezzo di comunicazione fra guidatore e veicolo, ha lo scopo di facilitare l'interazione fra i due agenti, nel suo sviluppo ci si è posti l'obiettivo di trovare un trade-off fra precisione e comfort.
- Politica di risoluzione dei conflitti: la risoluzione dei conflitti è essenziale per poter sfruttare a pieno la cooperazione fra guidatore e veicolo intelligente.

Nei prossimi paragrafi verranno analizzate nel dettaglio queste componenti.

2.2 Catalogo delle manovre

La prima parte del lavoro nello sviluppo del Digital Driving è stato impegnato nello sviluppo di un catalogo di possibili manovre che potesse soddisfare tutte le condizioni di guida. Per fare questo si sono analizzati i dati ricavati dal progetto *Vislab Intercontinental Autonomous Challenge* (VIAC) [28] [29].

Il laboratorio VisLab, dell'Università degli Studi di Parma, ha sviluppato negli anni diversi prototipi di veicoli autonomi o altamente automatizzati, tra cui ARGO [30], TerraMax [12, 11] e BRAiVE [31, 32]. Braive è l'ultimo prototipo sviluppato dal laboratorio ed è stato testato in modalità completamente automatica in diverse occasioni, sia all'interno che all'esterno del campus di Parma e anche in altre città. Le prestazioni di BraiVe sono risultate soddisfacenti nei test prodotti, ma questi test furono tutti organizzati in scenari strutturati e condizioni atmosferiche note. Ulteriori test con le stesse condizioni furono considerati poco vantaggiosi poiché avrebbero

aggiunto pochi risultati rispetto a quelli già ottenuti in laboratorio. Per questo motivo, VisLab ha deciso di organizzare un esperimento estensivo (più di due mesi di test continuativo), che comprendesse scenari differenti e che coinvolgesse continuamente la guida su strade senza limitazioni di accesso. Il test, il cui percorso è raffigurato in



Figura 2.3: Percorso del test VIAC.

figura 2.3, è stato svolto attraversando diversi paesi e testando un subset delle capacità automatiche di Braive. Oltre a questo, lungo tutto il percorso i dati provenienti dai sensori montati sui veicoli sono stati memorizzati ed etichettati per facilitarne la successiva visualizzazione ed analisi. Il viaggio è iniziato il 20 Luglio 2010, ha richiesto circa 3 mesi per una distanza totale di circa 13000 km; i veicoli automatici hanno viaggiato dall'Italia alla Cina attraverso Slovenia, Croazia, Serbia, Ungheria, Ucraina, Russia e Kazakhstan. Durante il viaggio le tecnologie impiegate sono state mostrate in diverse conferenze stampa e dimostrazioni live.

L'analisi dei dati ottenuti durante VIAC ha portato a diverse considerazioni ri-

guardo alla struttura delle manovre che devono essere implementate:

- Dimensioni del catalogo: per descrivere correttamente tutte le situazioni di guida è necessario implementare più di 20 manovre diverse. Questo porta a due problemi: l'implementazione dell'intero catalogo richiederà molto tempo e sarà necessario trovare un modo per ridurre il catalogo da presentare all'utente, poiché un catalogo così grande sarebbe difficile da imparare ed utilizzare.
- Fusione informazioni eterogenee: a questo livello di pianificazione è necessario unire informazioni diverse, quali posizione e velocità.
- Mantenimento dell'architettura a 3 livelli: l'architettura a 3 livelli che vedremo nel prossimo capitolo è utilizzabile anche per questo catalogo di manovre, di conseguenza il pianificatore a livello di manovra dovrà completare ma non sovrapporsi al pianificatore a livello di controllo.
- Ostacoli: a questo livello non devono essere considerati gli ostacoli, né fissi né in movimento, presenti sul percorso, sarà il pianificatore di basso livello ad occuparsene.

La risoluzione del primo problema è stata effettuata tramite una duplice via: in primo luogo si è creata una struttura standard di manovra, che vedremo in questa sezione, che offre una descrizione generica di una manovra ideale. Questo accorcia i tempi di sviluppo poiché per ogni manovra è sufficiente completare i campi parametrici della struttura generica; inoltre la struttura standard aiuta ad astrarre la manovra al corretto livello. In secondo luogo si è deciso di presentare all'utente un catalogo delle manovre più compatto di quello descritto, in modo che sia di più facile gestione.

La struttura base di manovra è costituita dai seguenti elementi:

- Pre-condizioni: lista di condizioni necessaria all'abilitazione della manovra. Se tutte le condizioni sono verificate la manovra è selezionabile. Le precondizioni servono per rendere sicura l'attivazione di una manovra;
- Post-Condizioni: lista di condizioni che causano il termine naturale della manovra. Queste condizioni servono a dare un limite temporale alla manovra;

- Condizioni di uscita: lista di condizioni che causa il termine immediato della manovra per questioni di sicurezza. Se una di queste condizioni viene verificata la manovra viene terminata immediatamente e il controllo passa ad un'altra manovra o al conducente se necessario;
- Gates: lista di *gates* che la manovra deve attraversare. Il concetto di *gate* verrà spiegato tra poco.

Il concetto di *gate* nasce per due motivi principali: creare una struttura standard utilizzabile in tutte le manovre ed unire informazioni eterogenee all'interno della stessa struttura. La sfida nella creazione della struttura dei *gates* è stata il definire una struttura che fosse abbastanza generica da poter descrivere ogni manovra e allo stesso tempo abbastanza descrittiva da poter essere tradotta direttamente in traiettoria dal pianificatore di livello di controllo.

Il concetto di *gate* si pone quindi come idea fondamentale di questo livello di pianificazione: un *gate* integra informazioni riguardo velocità, posizione e orientazione che il veicolo dovrà avere nella traiettoria futura senza dover implementare direttamente una traiettoria completa. Fondamentalmente, un *gate* è una sorta di *checkpoint* pianificato nella traiettoria futura che il veicolo deve attraversare con una certa orientazione ed una certa velocità. In particolare, ogni *gate* è costituito dai seguenti elementi:

- Posizione e orientazione: x , y e α del *gate* in coordinate mondo. Un aspetto chiave è che il veicolo deve passare all'interno del *gate* con la stessa orientazione del *gate* stesso.
- Dimensione: la larghezza identifica il massimo errore permesso nel passaggio attraverso il *gate*.
- Velocità massima: massima velocità permessa di passaggio, questo limite è imposto dal tipo di strada che si sta percorrendo o da qualche limite locale (opportuno segnalato da un cartello stradale o riportato sulla mappa).
- Velocità suggerita: velocità suggerita di attraversamento del *gate*. Questa velocità è utilizzata dal pianificatore al livello di controllo per determinare l'effett-

tiva velocità del veicolo (in assenza di ostacoli). Questo parametro è usato dal pianificatore a livello di manovra per cambiare l'effettiva velocità del veicolo.

La lista di gates è determinata da un pianificatore, che determina la posizione dei vari gates posizionandoli a distanza costante lungo la traiettoria. Ogni manovra ha un diverso pianificatore, ma tutti i pianificatori condividono tre dati: distanza di pianificazione, massima differenza di velocità e massima differenza di orientazione. La distanza di pianificazione identifica la distanza fra un gate ed il successivo, la massima differenza di velocità ed orientazione identificano le massime differenze di velocità e orientazione rispettivamente fra due gates consecutivi. Questi due ultimi parametri sono comunicati dal pianificatore a livello di controllo poiché sono dei limiti imposti dal pianificatore e dalla dinamica del veicolo intelligente (dati che sono a disposizione del solo livello di controllo).

Manovra	Descrizione
LKS+ACC	Sistema di mantenimento della carreggiata e della distanza dal veicolo che precede. Viene utilizzato in tutte le situazioni di guida su strada in cui sono presenti le linee di demarcazione.
Fuoristrada	Sistema di guida fuoristrada, viene utilizzato quando il veicolo si trova ad affrontare spazi non delimitati da strutture stradali.
Tienibordo	Sistema di guida basato sulla posizione dei limiti della strada, ad esempio fossati o alti marciapiedi; viene utilizzato in caso di navigazione in ambienti che prevedono questo tipo di delimitatori.
Inseguimento muro	Sistema di guida basato sulla posizione di elementi fissi a bordo strada, quali muri o guard-rail; questa manovra è utile in ambiente urbano dove ci si può trovare ad affrontare spazi stretti fra edifici molto alti.

Inseguimento GPS	Sistema di guida basato su una serie di punti GPS da inseguire; la lista di punti può essere fissa (se si dispone di una mappa molto precisa) oppure può essere trasmessa in tempo reale da un'altra sorgente, come ad esempio un veicolo che precede. Questa manovra risulta utile in caso di inseguimento di un altro veicolo.
Navigazione in spazio aperto	Sistema di guida che sfrutta la percezione del terreno in fronte al veicolo per determinare qual'è l'area percorribile; questo modulo non considera gli ostacoli ed è una manovra fondamentale che può essere sempre abilitata in caso di emergenza.
Sorpasso	Manovra di sorpasso, è unica per tutte le situazioni stradali ad almeno una corsia per senso di marcia.
Cambio corsia	Manovra utilizzata in caso di cambio corsia, sia destro che sinistro, in ambienti a più corsie per senso di marcia, come ad esempio quelli autostradali.
Inseguimento Veicolo	Manovra di inseguimento del veicolo che precede, utile in caso di situazioni di coda in ambiente stradale o per compiti molto dedicati come il platooning.
Rotonde	Manovra di gestione degli incroci di tipo rotonda; data la particolarità della navigazione all'interno di incroci di questo tipo si è dovuto creare una manovra apposita indipendente.
Incroci a T	Manovra di gestione degli incroci con intersezione generica a T, è utilizzato per la gestione di tutti i tipi di intersezione a T, indipendentemente dal numero di corsie.
Ingresso/uscita	Manovra di gestione degli ingressi uscita di strade a carreggiate separate, è utilizzata in caso di autostrade, tangenziali ed in generale in tutte le corsie di immissione/emissione da una strada.

Semaforo	Manovra di gestione delle intersezioni semaforiche, in tutte le loro varianti.
Parcheggio con angolo	Manovra di gestione dei parcheggi di tipo "a lisca di pesce".
Parcheggio parallelo	Manovra di gestione del parcheggio con posizionamento iniziale in parallelo rispetto al posto auto.

Tabella 2.1: Manovre derivanti dall'analisi dei dati dal progetto VIAC.

In tabella 2.1 sono elencate tutte le manovre estratte dall'analisi dei dati del progetto VIAC, con una breve descrizione delle caratteristiche e dei casi di uso principali di ogni manovra; di seguito vengono esemplificate solo alcune di queste manovre, per mostrare la struttura dei pianificatori a livello di manovra.

2.2.1 LKS+ACC

Lo scopo della manovra "LKS+ACC" è quello di permettere la guida all'interno di strade urbane od extraurbane, ad una o più corsie per senso di marcia ove siano presenti le linee di separazione della corsia/carreggiata modificando la propria velocità in modo da adattarla a quella dei veicoli che precedono. Questa manovra è quindi la manovra base che viene utilizzata in tutti i casi di navigazione su strada, ad eccezione di situazioni particolari, come quelle di un parcheggio, dove le linee di demarcazione non sono presenti.

La lista delle precondizioni di questa manovra è costituita da un solo elemento:

- Presenza delle demarcazioni di corsia.

Questa è infatti l'unico elemento necessario per l'attivazione della manovra; inoltre la manovra non ha post-condizioni in quanto non ha un termine naturale ma dura finchè le delimitazioni di corsia sono presenti. L'elenco delle condizioni di uscita è anch'esso composto da un solo elemento:

- Mancanza dei demarcatori di corsia

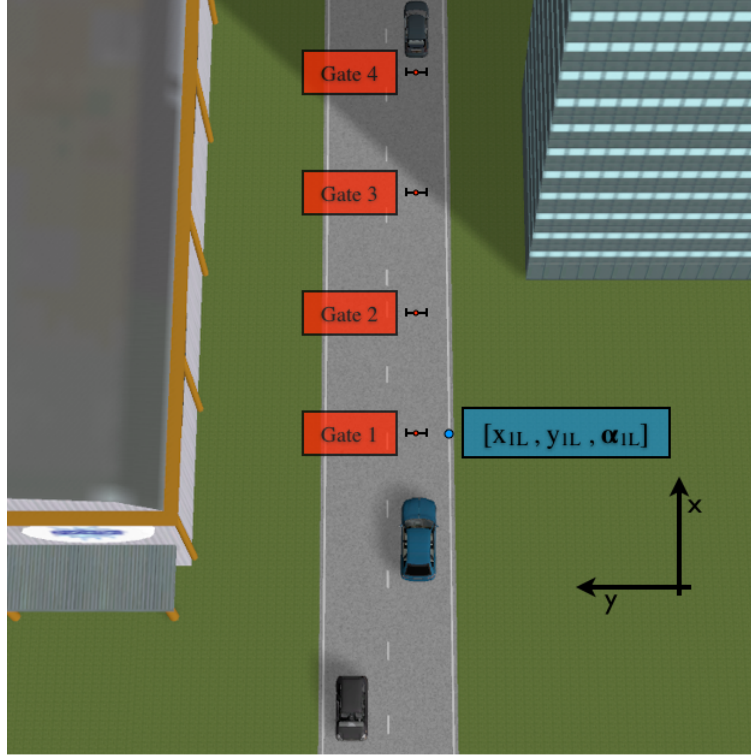


Figura 2.4: Struttura della lista di gates per la manovra "LKS+ACC".

In figura 2.4 è riportata la lista dei gate per una manovra di "LKS+ACC"; com'è possibile vedere ogni gate è posizionato a spazio costante parallelamente alla direzione della corsia in cui il veicolo si trova. In particolare i dati di ogni gate sono calcolati come in equazione 2.1.

$$\begin{aligned}
 \mathbf{x}_k &= \mathbf{x}_L \\
 \forall k \in [0, N], \quad \mathbf{y}_k &= (\mathbf{y}_L - \mathbf{y}_r)/2 \\
 \alpha_k &= (\alpha_L - \alpha_R)/2
 \end{aligned} \tag{2.1}$$

La metodologia di assegnamento del profilo di velocità è descritta nell'algoritmo 1. In questo caso V_f è la velocità suggerita al gate finale, V_{nv} è la velocità del veicolo che precede ed N il numero di gates da pianificare.

Algorithm 1 Profilo di velocità in LKS+ACC

```

1: if ( $MaxSpeed - V_{nv}$ ) >  $th$  then
2:    $V_f = (V_{nv} + MaxSpeed)/2$ 
3: else
4:    $V_f = V_{nv}$ 
5: end if
6:  $\delta_k = (v_f - v_i)/N$ 
7: for all gates  $k \in [0, N]$  do
8:    $v_k = v_{k-1} + \delta_k$ 
9: end for
  
```

2.2.2 Sorpasso

Lo scopo della manovra chiamata "Sorpasso" è quello di gestire l'operazione di sorpasso in tutte le situazioni stradali; in sintesi la manovra si occupa di spostarsi nella corsia sinistra e sorpassare il veicolo che precede senza preoccuparsi di tornare nella corsia di destra poiché all'interno dell'architettura, come vedremo più tardi, è sempre attiva una modalità di ritorno automatico nella corsia più a destra (a meno di limiti locali), come previsto dalla attuale legislazione europea. In questa sezione viene mostrato il pianificatore per la sola situazione di guida a più corsie per senso di marcia.

La lista delle precondizioni per la manovra di "Sorpasso" è costituita dai seguenti elementi, il cui significato può essere ricostruito anche dalla figura 2.5:

- Rilevazione corretta delle linee di demarcazione stradale: le linee di demarcazione sono necessarie per il calcolo dei gate, come vedremo in seguito;

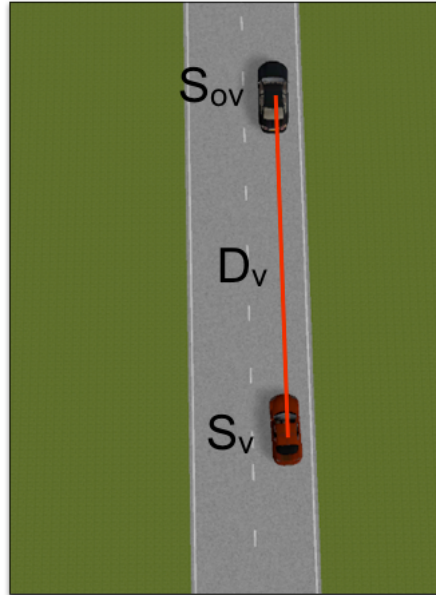


Figura 2.5: Attivazione della manovra di sorpasso.

- $S_v < S_{sugg} + th$, dove S_v è la velocità attuale del veicolo intelligente, S_{sugg} è la velocità suggerita dalla precedente manovra e th un valore di soglia. Se l'espressione è verificata significa che la velocità del veicolo è fortemente limitata da un altro veicolo che precede e di conseguenza è consigliabile attivare una manovra di sorpasso;
- Corsia sinistra libera: questa informazione viene fornita da una applicazione di percezione dedicata (Blind Spot Monitor).

Come detto in precedenza, il verificarsi contemporaneo dell'intera lista di precondizioni causa l'attivazione della manovra di sorpasso. La lista delle post-condizioni è costituita dai seguenti elementi:

- $D_v < -L + th$, dove L è la lunghezza del veicolo intelligente, th una soglia e D_v la distanza fra i due veicoli. Questa espressione codifica l'effettivo avvenuto

sorpasso fra i due veicoli;

- $S_v > S_{ov}$ dove S_{ov} è la velocità del veicolo da superare. Questa condizione serve ad evitare eventuali situazioni di stallo ed a permettere al veicolo intelligente un ritorno veloce nella corsia di destra.

Infine la lista delle condizioni di uscita è la seguente:

- $S_{ov} > MaxVel$: se la velocità del veicolo da superare (S_{ov}) diventa maggiore della velocità massima del tratto di strada che si sta percorrendo la manovra termina immediatamente in quanto il veicolo non si trova più nelle condizioni di sicurezza necessarie per effettuare la manovra stessa.

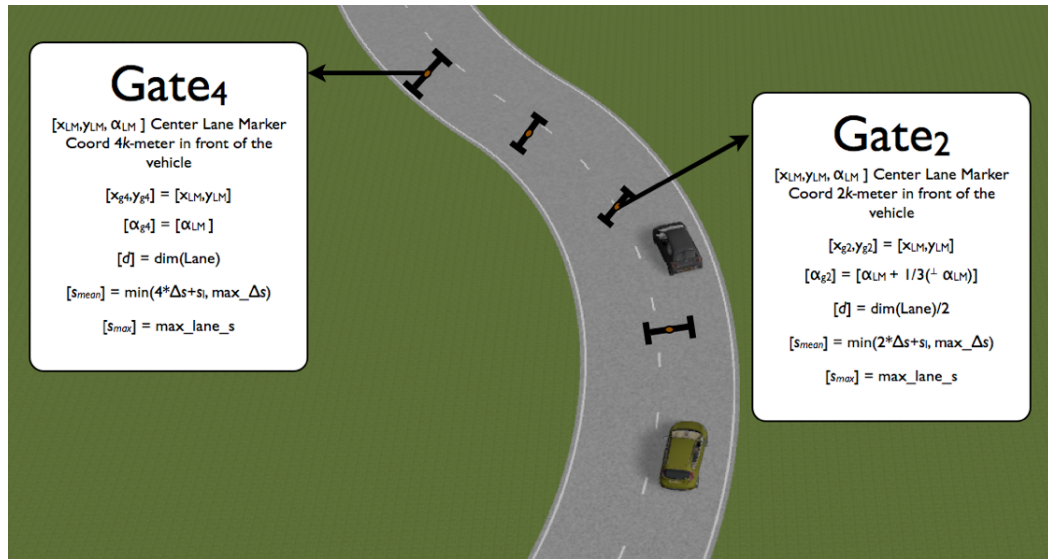


Figura 2.6: Sorpasso: lista dei gates.

L'elemento di maggiore importanza all'interno della manovra è sicuramente la lista dei gates, un esempio di come essa è determinata per la manovra di "Sorpasso" è raffigurato in figura 2.6. Inizialmente vengono presi due punti di riferimento, entrambi

basati sulla linea di demarcazione che separa le corsia destra dalla corsia sinistra (P_c e P_f in figura 2.6). P_c è il punto in cui il veicolo attraversa la linea di demarcazione centrale e P_f è il punto in cui il veicolo sarà completamente nella corsia di sinistra. $[x, y, \alpha]$ dei 4 gates calcolati al primo step sono derivati direttamente da questi punti di riferimento tramite le equazioni descritte in figura 2.6. Da sottolineare come la traiettoria non debba essere calcolata rigorosamente, tenendo conto per esempio della posizione del veicolo da superare, poiché tutti questi dati sono a disposizione del pianificatore a livello di controllo che si occuperà di calcolare la traiettoria reale con annesso *avoiding* degli ostacoli.

Per quanto riguarda il profilo di velocità esso è ricavato ponendo la velocità dell'ultimo gate come velocità massima raggiungibile ed interpolando gli altri dati come descritto nell'equazione 2.2.

$$\begin{aligned} \delta_v &= \min((\mathbf{v}_f - \mathbf{v}_i)/N, \mathbf{MaxSpeed}) \\ \forall k \in [0, N], \mathbf{v}_k &= \mathbf{v}_i + k\delta_v \end{aligned} \quad (2.2)$$

Quella descritta fino ad ora è la pianificazione iniziale della manovra di sorpasso, ad ogni step successivo di pianificazione (a tempo costante) la posizione dei gate ancora da varcare viene rifinita utilizzando i nuovi dati sulla percezione (in particolare in riferimento alla posizione dei demarcatori di carreggiata); infine nuovi gates vengono aggiunti nella corsia di sinistra in accordo con la manovra di "LKS+ACC" descritta nel paragrafo 2.2.1.

2.2.3 Ingresso/Uscita

La manovra "Ingresso/Uscita" è utilizzabile in tutti i casi in cui sia necessario entrare/uscire da una strada con immissione/uscita effettuabile tramite una corsia di accelerazione/decelerazione apposita. La corsia di accelerazione/decelerazione è una corsia specializzata per consentire ed agevolare l'ingresso/uscita ai veicoli sulla carreggiata, è a senso unico e può essere costituita da una o più corsie. In questo para-

grafo viene esemplificato il caso di immissione in una strada a due corsie, come ad esempio una autostrada; tutti gli altri casi sono simili e ricavabili direttamente da quanto presentato in questo paragrafo. La lista delle precondizioni di attivazione di questa manovra è costituita dai seguenti elementi:

- Necessità di ingresso nella strada, comunicata dal pianificatore a livello di navigazione o dal guidatore;
- Presenza di uno svincolo di ingresso: rilevato tramite la percezione ed estratto dalla mappa cartografica;
- Presenza delle linee di demarcazione della carreggiata.

In sintesi, la manovra è abilitata quando si ha la volontà di cambiare strada, entrando (in questo caso) nello svincolo di immissione di una strada a più corsie per senso di marcia. La lista delle post-condizioni è costituita dai seguenti elementi:

- Fine della corsia di immissione
- $v_f > 0$

Considero di conseguenza terminata la manovra quando, dopo essermi spostato nella corsia di sinistra, non percepisco più la corsia di accelerazione, inoltre impongo che la velocità del veicolo sia > 0 in modo da evitare situazioni di stallo. L'unica condizione di uscita presente è la mancata percezione dei delimitatori di carreggiata, che renderebbe impossibile la corretta immissione del veicolo all'interno della propria corsia. In figura 2.7 è rappresentato un esempio di lista di gates per la manovra di immissione; la posizione dei gates è calcolata indipendentemente dallo stato della corsia di sinistra, questa informazione infatti influenza solamente il profilo di velocità. Dopo un primo in cui il pianificatore calcola la posizione dei gates esattamente come visto nel paragrafo 2.2.1 per la manovra di "LKS+ACC" si arriva nella corsia di accelerazione in cui la linea di carreggiata sinistra risulta a tratti. A questo punti vengono fissati due punti di riferimento: P_c , cioè il punto in cui viene attraversata la linea di demarcazione sinistra e P_f , cioè il punto in cui il veicolo si trova completamente entro la strada di cui sta percorrendo l'immissione. Definiti questi punti

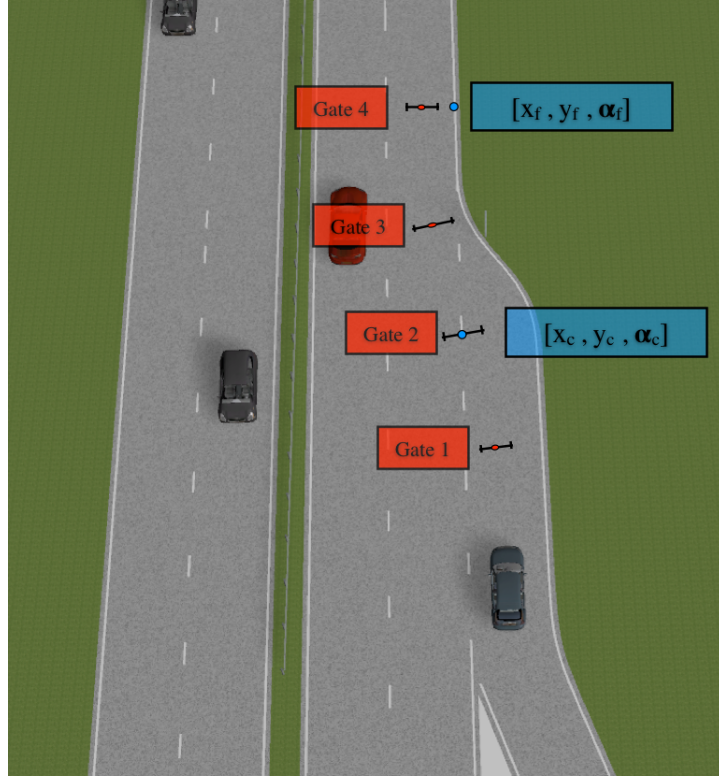


Figura 2.7: Ingresso/Uscita: lista dei gates.

vengono calcolati i dati dei gate detti "Gate2" e "Gate4" in figura 2.7 nel seguente modo:

$$[x_2, y_2, \alpha_2] = [x_c, y_c, (4/3)\alpha_c] \quad (2.3)$$

$$[x_4, y_4, \alpha_4] = [x_f, (y_{fL} - y_{fR})/2, \alpha_f] \quad (2.4)$$

Una volta determinati i sopracitati gates si possono ottenere i valori di Gate1 e Gate3 di figura 2.7 interpolando linearmente i dati dei gates appena trovati.

Per quanto riguarda il profilo di velocità esso viene calcolato come mostrato nell'algoritmo 2. Dall'analisi dell'algoritmo si evince che il pianificatore, se la corsia

Algorithm 2 Profilo di velocità in LKS+ACC

```

1: if Corsia sinistra libera then
2:    $\delta_v = \min((MaxSpeed - v)/N, Max\delta_{Speed})$ 
3: else
4:    $\delta_v = \min((-v)/N, MaxNegative\delta_{Speed})$ 
5: end if
6:
7: for all gates  $k \in [0, N]$  do
8:    $v_k = v_{k-1} + \delta_k$ 
9: end for
  
```

sinistra risulta libera, cerca di accelerare per raggiungere la corsia sinistra con la velocità desiderata, viceversa se la corsia risulta occupata cerca di frenare in modo da non invadere la corsia; le differenze di velocità fra i vari gates, espressa dalla variabile δ_v nell'algoritmo 2 viene confrontata con la massima differenza di velocità sostenibile dal pianificatore a livello di controllo, in modo da evitare configurazioni con velocità non raggiungibili dal veicolo. La logica appena presentata che calcola il profilo di velocità è valida solamente per la fase di attraversamento della linea di demarcazione ed è stata presentata in quanto è la più interessante dal punto di vista della pianificazione, in realtà essa è preceduta e seguita da una fase in cui sia i gates che il profilo della velocità è calcolato esattamente come per la manovra di "LKS+ACC" del paragrafo 2.2.1.

2.3 Behaviours

Parte del lavoro effettuato per il Digital Driving è stato impiegato nel rendere il catalogo il più completo e compatto possibile. La completezza del catalogo è stata raggiunta partendo dall'analisi di un set di dati molto grande come quelli ricavati dal progetto VIAC. Nel paradigma di guida a livello di manovra il guidatore deve cono-

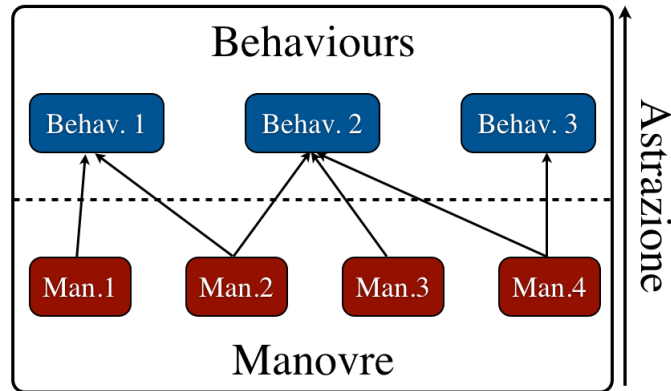


Figura 2.8: Rapporto fra behaviours e manovre nel Digital Driving.

scere perfettamente l'intero catalogo delle manovre per poter controllare il veicolo, di conseguenza il numero di manovre di cui è composto il catalogo diventa uno dei fattori chiave per la riuscita del paradigma, come dimostrato in [33]. In letteratura sono stati adottati diversi metodi per contenere le dimensioni del catalogo delle manovre: in [21] viene utilizzato un approccio analitico al problema che distingue più di 3000 di possibili manovre, mentre in [33] viene utilizzato un approccio più semplice costituito da 17 manovre; quest'ultimo catalogo delle manovre è quello utilizzato all'interno di Conduct-by-Wire.

L'approccio che è stato utilizzato all'interno di questa tesi di dottorato è totalmente differente: per riuscire a ridurre il numero delle manovre pur mantenendo la completezza evidenziata nel capitolo precedente si è deciso di suddividere orizzontalmente il catalogo in due diversi livelli. Il layer più alto, quello esposto all'utente, è costituito da obiettivi di alto livello chiamati "Behaviours" mentre il secondo e più basso livello è costituito dalle manovre utili per raggiungere l'obiettivo, cioè dalle manovre viste sino a questo momento.

Considerando ad esempio la guida in ambiente autostradale risulta evidente come l'obiettivo finale è in realtà legato dal modo in cui esso viene raggiunto: la guida in autostrada può essere implementata infatti tramite un sistema di tienicorsia, oppure

un inseguitore di veicolo oppure potrei localmente utilizzare una manovra di sorpasso. Tutte le manovre qui elencate non cambiano però l'obiettivo finale che è e rimane la guida in ambiente autostradale. All'interno del Digital Driving quindi il guidatore dovrà solamente selezionare l'obiettivo della manovra che vuole compiere, nel caso in esempio "Guida Autostradale", e sarà poi la struttura stessa del behaviour a decidere quale manovra effettivamente verrà attivata fra quelle disponibili. Il rapporto fra

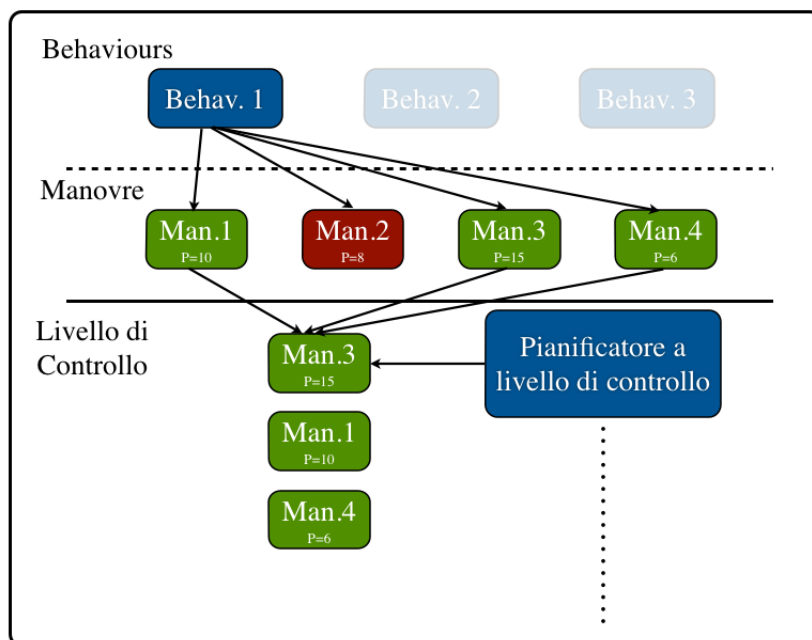


Figura 2.9: Meccanismo di abilitazione dei behaviours e delle manovre.

le manovre ed i behaviours è di tipo multi-a-molti, come evidenziato in figura 2.8; infatti ad un behaviour appartengono più manovre ma una stessa manovra appartiene a behaviours diversi. Inoltre all'interno di ogni behaviour le manovre sono ordinate attraverso un valore di priorità e la scelta di quale manovra deve essere eseguita in ogni istante avviene proprio attraverso questo valore. Fondamentalmente, come mostrato in figura 2.9, vengono valutate le precondizioni di ogni manovra componente

il behaviour: se sono tutte verificate la manovra viene considerata abilitata. Ripetendo il processo per ogni manovra se ne ottiene una lista di manovre disponibili per il behaviour in questione; il pianificatore di ogni manovra viene quindi eseguito e la lista dei gates calcolata per ciascuna manovra viene inviata al pianificatore a livello di controllo, ordinata per valore di priorità. Il pianificatore a livello di controllo cercherà di interpolare una traiettoria in ogni lista di gates finché non ne troverà una sicura e fattibile. Risulta necessario infatti ricordare che il pianificatore a livello di manovra non considera né i possibili ostacoli presenti nello scenario né la dinamica del veicolo. Se nessuna delle manovre passa il test delle precondizioni l'intero behaviour viene disabilitato poiché è considerato non sicuro o non adatto alla particolare situazione di guida; i behaviours disabilitati non vengono proposti come opzioni di guida al conducente.

In tabella 2.2 sono riportati i behaviours che compongono il catalogo del Digital Driving e le manovre che costituiscono ognuno di essi. Il numero riportato nella stessa tabella è la priorità che la manovra ha all'interno del behaviour. La divisione del livello di manovra in due sottolivelli e il conseguente raggruppamento delle manovre in behaviours ha prodotto un catalogo completo e compatto; inoltre permette un grande riuso delle componenti sviluppate grazie al rapporto molti-molti che esiste fra i behaviours e le manovre.

Manovra	Urban Mode	Highway Mode	Vehicle Fol-lower	Stop&Go	CrossRoad Selection	Parking
LKS+ACC	5	5				
Fuoristrada	5					
Tienibordo	3	3				
Inseguimento Muro	3					
Inseguimento GPS	3		3			
Sorpasso	8	10	10			
Navigazione in spazio aperto	1	1	1	1	1	1
Cambio Corsia		8	8			
Inseguimento veicolo			5	5		
Rotonde					5	
Incroci a T					5	
Ingresso/Uscita					5	
Semaforo	10				5	
Parcheggio con angolo						10
Parcheggio parallelo						15

Tabella 2.2: Composizione dei behaviours, il numero in tabella corrisponde alla priorità che la manovra ha all'interno del behaviour.

Di seguito una breve descrizione esplicativa delle finalità e delle situazioni di utilizzo di ogni behaviour di tabella 2.2:

- Urban Mode: questo behaviour è concepito per essere usato in ambiente urbano o in fuoristrada. Comprende la possibilità di percorrere strade a singola corsia o fuoristrada, di sorpassare veicoli e di controllare e rispettare gli incroci semaforici e le rotonde;
- Highway Mode: behaviour specializzato nella guida in strade a più corsie per senso di marcia, permette di effettuare, oltre alla normale guida, sorpassi e di cambiare corsia. Per come è costituito il veicolo rimane sempre nella corsia più a destra libera;
- Vehicle Follower: behaviour specializzato nell'implementare il paradigma del *Leader-Follower* [34]. In questo paradigma un veicolo intelligente (Follower) può inseguire, anche a distanza, un veicolo intelligente detto Leader osservando nel frattempo le regole del traffico ed evitando gli ostacoli sul percorso;
- Stop&Go: specializzato nella gestione delle code, questo behaviour permette di seguire a brevissima distanza ed a velocità molto bassa un veicolo che precede, ignorando la segnaletica orizzontale se necessario;
- CrossRoad Selection: behaviour di gestione di tutti i tipi di incrocio, permette di cambiare strada indipendentemente dal tipo di strada/incrocio;
- Parking: behaviour di parcheggio, permette non solo di parcheggiare il veicolo ma di navigare all'interno di un parcheggio rispettando le regole della strada finché non viene trovato automaticamente un posteggio libero.

Per concludere questa sezione si porta ora un esempio di una situazione di guida, per chiarire come il meccanismo di scambio delle manovre sia in effetti efficace. Nel caso raffigurato in figura 2.10 si ha una situazione di guida autostradale in cui il veicolo intelligente non ha nessun veicolo che lo ostacoli, inoltre la corsia sinistra risulta occupata. Per questo motivo la manovra "Sorpasso" risulta disabilitata, poiché alcune delle sue precondizioni non sono rispettate. Per lo stesso motivo è disabilitata

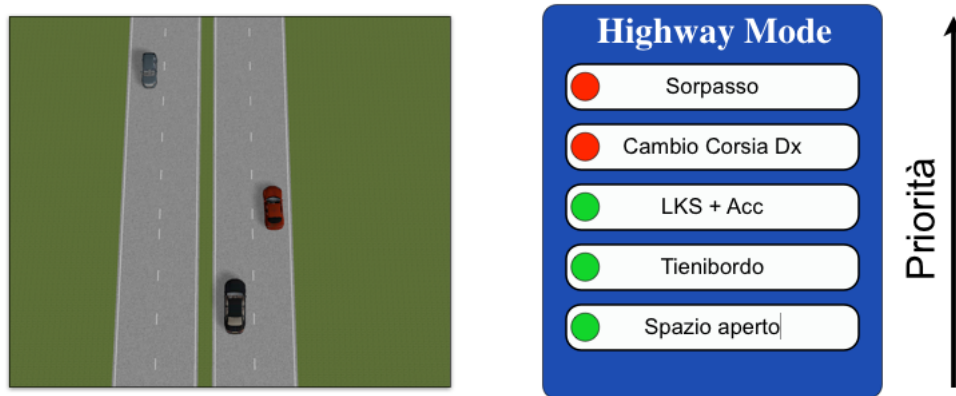


Figura 2.10: Utilizzo della priorità in *Highway Mode*.

la manovra "Cambio Corsia Destro", poiché il veicolo si trova già nella corsia più a destra. Le manovre con priorità minore, a partire da "LKS+ACC" fino a "Navigazione in spazio libero" possono essere invece tutte attivate contemporaneamente poiché le loro condizioni sono tutte verificabili; di conseguenza la lista dei gates prodotta da ogni manovra abilitata viene trasmessa al pianificatore a livello di controllo, come già raffigurato in 2.9.

2.4 Risoluzione dei conflitti

Come visto nel paragrafo 1.3 i conflitti che nascono a livello di Guida Altamente Automatizzata sono tra i più critici, principalmente per due motivi:

- Responsabilità nella guida: a questo livello guidatore e veicolo intelligente hanno all'incirca lo stesso grado di responsabilità nella guida del veicolo, di conseguenza non è chiaro a chi spetti la competenza di risoluzione dei problemi;
- Indiscriminazione: a questo livello la percezione e l'autonomia del sistema intelligente sono molto avanzate ed, in caso di differente interpretazione del-

le realtà fra guidatore e veicolo risulta impossibile capire se si tratta di una imprudenza umana o di un errore di valutazione del software.

La soluzione proposta all'interno di questa tesi consiste nella risoluzione multi-livello dei conflitti, in modo da risolvere ogni problema agendo allo stesso livello di astrazione in cui esso nasce. Più precisamente, i conflitti sono risolti in tre modi:

- Sicurezza dell'intelligenza artificiale: tramite il meccanismo delle precondizioni, il veicolo intelligente propone al guidatore solamente manovre che ritiene sicure. Di conseguenza il veicolo non può essere forzato a compiere manovre che ritiene insicure da parte dell'utente;
- Predominanza del guidatore: il guidatore, in quanto responsabile del veicolo, deve avere predominanza nelle situazioni critiche sul veicolo intelligente: per questo motivi gli è lasciata la possibilità, in qualsiasi momento, di utilizzare i classici attuatori per sovrascrivere gli ordini a livello di controllo. Ad esempio, se volesse superare un veicolo anche se l'intelligenza artificiale non lo ritiene sicuro, il guidatore può prendersi la responsabilità di impugnare il volante ed effettuare manualmente il sorpasso.
- Cooperazione: al guidatore è lasciata la possibilità di cooperare con l'intelligenza artificiale anche a livelli più alti: per esempio può utilizzare la HMI del Digital Driving per cambiare la pianificazione effettuata a livello di navigazione in modo da forzare un percorso diverso da quello originariamente pianificato. Allo stesso tempo il veicolo intelligente può far affidamento sul guidatore in caso di decisioni critiche, ponendo scelte al guidatore stesso tramite lo schermo interno all'HMI del Digital Driving.

La predominanza del conducente risulta essere anche un prerequisito di legge: come evidenziato nell'appendice A, un veicolo intelligente deve sempre riservare l'opzione di sovrascrittura dei comandi al conducente per poter essere consistente con l'attuale regolamentazione della Convenzione di Vienna.

Capitolo 3

Architettura software basata su diversi livelli di automazione

Il genio è un uomo capace di dire cose profonde in modo semplice.

– Charles Bukowski

3.1 Open intelligent Architecture for Future Autonomous Vehicle

Il progetto "Open intelligent Architecture for Future Autonomous Vehicle" (OFAV), di cui questa tesi è uno dei risultati, è un progetto dell'Unione Europea, iniziato nel 2008 presso il VisLab dell'Università degli Studi di Parma. L'obiettivo del progetto è lo sviluppo di una architettura aperta per i veicoli intelligenti della prossima generazione che possa diventare uno standard condiviso da tutti i produttori di automobili. Il progetto è basato su una suite sensoriale che include una percezione a 360° e su dei moduli di decisione e pianificazione con l'obiettivo ultimo di implementare un veicolo autonomo e/o con capacità di guida altamente automatizzate. I vari moduli

di percezione potranno contenere meccanismi di comunicazione vehicle-to-vehicle (V2V) e vehicle-to-infrastructure (V2I) per incrementare le capacità di percezione del veicolo intelligente. Questo progetto è stato assegnato a VisLab per l'esperienza che questo laboratorio ha nell'allestimento di veicoli automatici in quanto VisLab è già stato protagonista di alcune *milestones* nel campo della robotica mobile e dei veicoli intelligenti.

Il progetto VIAC, descritto brevemente nel paragrafo 2.2, ha rappresentato la prima grande milestone del progetto OFAV in quanto è stato il primo esempio di allestimento da zero di un veicolo elettrico completamente automatico; inoltre è stata l'occasione per un grande test delle componenti software ma soprattutto delle componenti hardware dei veicoli allestiti [35] [36].

Durante lo sviluppo del progetto VIAC ho potuto assistere e partecipare personalmente all'allestimento di un veicolo autonomo e ne ho ricavato una lista di problematiche alle quali si deve far fronte in questo tipo di attività. Banalmente, il problema principale da affrontare è la complessità del compito richiesto, che si manifesta principalmente nelle seguenti caratteristiche:

- **Competenze:** per realizzare un veicolo autonomo sono necessarie competenze che spaziano dalla meccanica, all'elettronica, all'intelligenza artificiale fino alla visione artificiale. Inoltre è necessario avere esperienza nell'uso dei numerosi sensori di cui è dotato, come telecamere, laser e antenne GPS.
- **Integrazione tra gruppi diversi:** una conseguenza diretta del punto sopra è il fatto che allo stesso progetto partecipano elementi e gruppi con *background* fortemente diversi ed è quindi necessario prendere degli accorgimenti in modo che tutte queste componenti possano collaborare adeguatamente all'interno del progetto.
- **Fluidità:** il carattere fortemente di ricerca di questo tipo di attività fa sì che il sistema tenda a cambiare di frequente, con il progressivo prendere coscienza, da parte del team, delle reali funzionalità richieste. L'architettura di sistema deve essere quindi progettata e costruita in modo da facilitare il più possibile l'ag-

giornamento o la sostituzione di ogni funzionalità, senza richiedere la modifica o la sostituzione di altre funzionalità.

- Debug: il *debug* di un veicolo autonomo è particolarmente critico, in quanto esso è generalmente un "pezzo unico", una singola risorsa alla quale tutti i componenti del gruppo necessitano di accedere. Raramente si ha a disposizione un secondo veicolo, clone del primo, su cui effettuare test in parallelo.
- Complessità dei Test: come detto prima, banalmente, uno dei problemi principali risulta essere la complessità generale del problema. Questo si riflette anche nella fase di test del sistema in quanto in caso di problemi, data la complessità della struttura, risulta difficile disaccoppiare le fonti di ogni problema. Risulta quindi necessario creare l'architettura in modo che ogni singolo pezzo sia disaccoppiabile e sperimentabile singolarmente, in modo da dover poi affrontare problemi relativa solo all'interconnessione fra le parti. Un simulatore di autoveicoli potrebbe essere una soluzione ottimale poiché permetterebbe, oltre al testing di un singolo elemento mentre tutti gli altri funzionano, di risolvere almeno parzialmente anche i problemi visti al punto precedente. Nell'appendice B è descritta la procedura di testing attraverso un simulatore utilizzata nell'analisi delle performance e nel debug dell'architettura descritta in questa tesi.

3.2 Stato dell'arte

Nel campo delle architetture per veicoli intelligenti grandi progressi sono stati fatti in occasione delle Challenge organizzate dal DARPA [37] [38] [39]; in occasioni di questi eventi infatti ogni team, dovendo allestire un veicolo automatico per una gara molto probante, ha posto molta attenzione nello sviluppo dell'architettura. Inoltre, spesso i team erano composti da organizzazioni molto eterogenee, provenienti sia dalla ricerca universitaria che da quella industriale e militare, e di conseguenza ci si è scontrati con i problemi di organizzazione e cooperazione elencati in precedenza.

Di seguito vengono presentati due architetture, costituenti lo stato dell'arte, utilizzate durante e dopo la Darpa's Urban Challenge.

3.2.1 Architettura del team Oshkosh

Il team Oshkosh ha partecipato sia alla Urban Challenge che alla Grand Challenge con un veicolo di grandi dimensioni chiamato TerraMax, una versione modificata di un veicolo tattico militare di medie dimensioni [11]. Il team era composto dalla Oshkosh Corporation, Teledyne Scientific and Imaging Company, VisLab dell'Università degli Studi di Parma, Ibeo Automotive Sensor GmbH, e Auburn University. Per la Urban Challenge, Oshkosh ha procurato il veicolo, e ha curato il management e le linee di progettazione generale per tutte le componenti hardware, software e di controllo; inoltre Oshkosh si è occupata di tutte le componenti elettriche e meccaniche, oltre alla progettazione e sviluppo degli algoritmi di controllo del veicolo di basso e medio livello. Teledyne si è occupata dell'architettura di sistema, della pianificazione della traiettoria e delle missioni; il VisLab si è occupato delle capacità percettive del veicolo, soprattutto dal punto di vista della visione artificiale. Ibeo Automotive Sensor GmbH ha sviluppato l'integrazione software dei vari LIDAR montati sul Terramax, mentre l'Università di Auburn si è occupato del GPS e dell'unità inerziale del veicolo.

La base dell'architettura software del Team Oshkosh è costituita da una architettura a livelli [40] composta da diversi moduli, detti servizi, che forniscono specifiche funzioni all'intero sistema; l'architettura è presentata in figura 3.1. Ogni modulo è descritto come un servizio che interagisce con gli altri tramite un set di messaggi asincroni. I tipi di servizio, come illustrato in figura 3.1, sono principalmente due: i servizi autonomi, i quali forniscono funzionalità per i comportamenti autonomi del veicolo, ed i servizi di sistema: di utilità generica, questi servizi sono di aiuto per l'affidabilità generale del veicolo e delle missioni. Di seguito una breve descrizione dei più importanti servizi della parte dei servizi autonomi:

- **Autonomous Vehicle Manager: (AVM)** gestisce tutte le operazioni di alto livello del veicolo, compresa la pianificazione. Questo servizio riceve i dati dalla percezione, che interpreta e trasforma in comportamenti adeguati del veicolo;

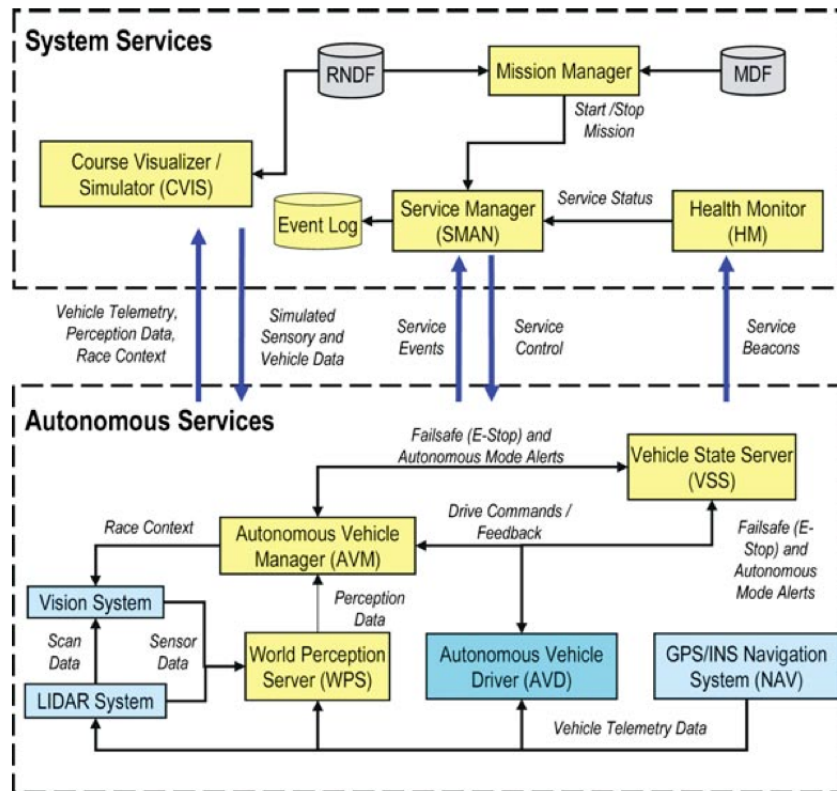


Figura 3.1: Architettura software del veicolo Terramax.

inoltre calcola la traiettoria più opportuna che invia all'Autonomous Vehicle Driver (AVD);

- **Autonomous Vehicle Driver: (AVD)** si occupa di tutte le operazioni di basso livello nel veicolo. AVD fornisce il controllo laterale, longitudinale e di stabilità ricevendo messaggi dal servizio AVM e traducendoli in attuazione tramite l'interfaccia Control-by-Wire;
- **World Perception Server:** questo servizio si occupa di raccogliere le informa-

zioni prodotte dai sensori, e di riorganizzarle nella forma di una rappresentazione organica e completa del mondo circostante il veicolo. I suoi compiti principali sono la raccolta di informazioni sensoriali dai sensori di percezione, la fusione sensoriale, cioè l'aggregazione dei dati raccolti dai vari sensori in una mappa complessiva del mondo circostante e infine la pubblicazione della mappa di percezione verso i servizi di pianificazione, sia in modalità *one-shot* che in modalità continuativa.

- Vision System: questo servizio pubblica i dati processati dalle varie applicazioni di visione. In particolare questi dati possono essere informazioni come la posizione delle corsie, o del percorso o infine la posizione degli ostacoli. Tutti questi dati sono inviati al WPS per l'aggregazione e la distribuzione agli altri servizi dell'architettura.

Riassumendo, l'architettura del veicolo Terramax è costituita da diversi moduli, disposti su diversi livelli, che basandosi sull'idea di servizio vogliono fornire funzionalità all'intero sistema attraverso la rete. Questo rende il sistema modulare e permette di disaccoppiare la posizione fisica del servizio dalla sua utilità funzionale, contribuendo a rendere il sistema scalabile.

3.2.2 Architettura del prototipo Junior

Junior [41], un prototipo basato su una Volkswagen Passat modificata, è stata sviluppata da un team di ricercatori provenienti principalmente dall'Università di Stanford (California, USA) e dalla Volkswagen, oltre che dai loro sponsor partner: Applanix, Google, Intel, Mohr Davidow Ventures, NXP, e Red Bull. Il team era composto principalmente dagli stessi elementi che vinsero la Darpa Grand Challenge nel 2005 [42]. Alla Urban Challenge Junior si è classificata seconda, dietro il veicolo vincitore della Carnegie Mellon.

L'architettura software di Junior è progettata come una pipeline basata sui dati in cui ogni modulo processa asincronamente le informazioni di cui è responsabile. Ogni modulo comunica con gli altri attraverso un protocollo di messaggi anonimi di tipo pubblicazione/sottoscrizione, basato sul "Inter Process Communication Toolkit"

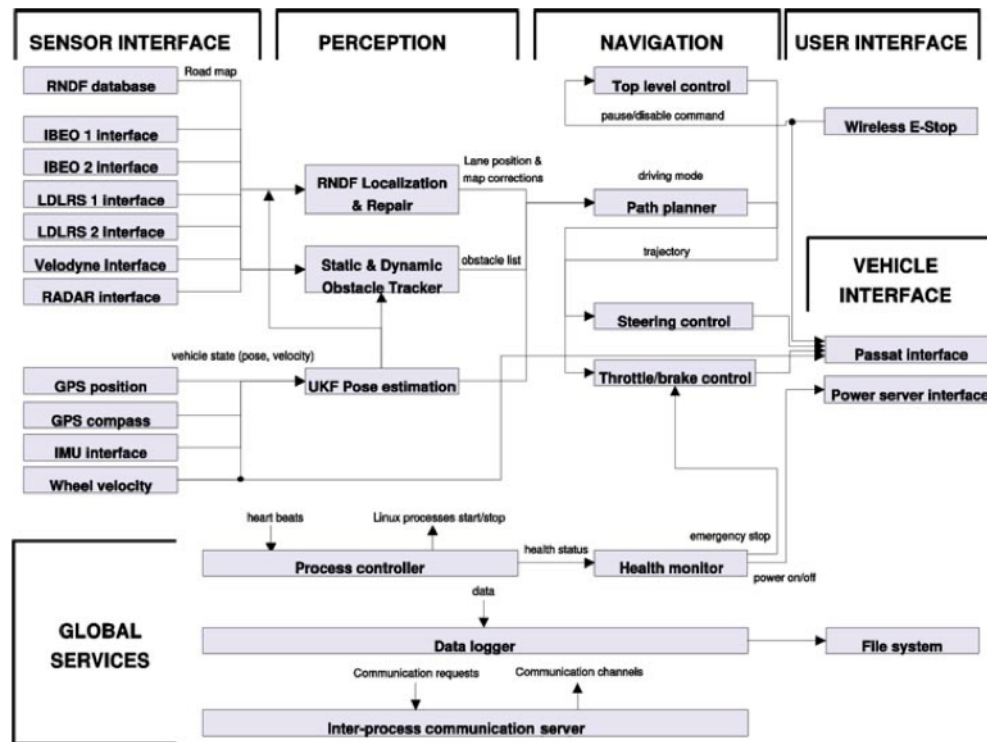


Figura 3.2: Architettura software del veicolo Junior.

(IPC). Ogni modulo sottoscrive la ricezione di messaggi da flussi mandati da altri moduli; i messaggi sono poi inviati asincronamente. Il risultato dell'elaborazione di un modulo può essere quindi pubblicato ed inviato ad altri moduli; in questo modo ogni modulo è sempre impegnato nell'elaborazione di un dato e l'intero sistema funziona come una pipeline. Il ritardo fra il tempo d'ingresso di un dato e il momento in cui esso influenza l'attuazione del veicolo è di circa 300 ms.

I vari moduli, il cui schema è rappresentato in figura 3.2, sono raggruppati in 5 gruppi:

- Interfaccia dei sensori: gestisce la comunicazione fra il veicolo ed i vari sensori

e rende disponibili al resto dei moduli software i dati prodotti dai sensori.

- Moduli di Percezione: segmentano l'ambiente in oggetti statici e dinamici, inoltre permettono una precisa localizzazione del veicolo relativamente alla mappa dell'ambiente.
- Moduli di Navigazione: determinano il comportamento effettivo del veicolo. Il gruppo di moduli di navigazione consiste in un certo numero di pianificatori unitamente ad una macchina a stati finiti che ha il compito di invocare i vari comportamenti dell'autoveicolo, prevenendo in questo modo situazioni di *deadlock*.
- Drive-by-Wire: abilita il controllo software di acceleratore, freno, sterzo, cambio, indicatori di direzione e freno d'emergenza.
- Servizi Globali: forniscono vari servizi a livelli di modulo, come il logging dei messaggi, timestamp, aiuto nel passaggio dei messaggi e funzioni di controllo del funzionamento dell'intera architettura.

La parte più interessante di questa architettura è costituita dal meccanismo di scambio messaggi che semplifica la ricerca della modularità e facilita la scalabilità del sistema; inoltre l'architettura a pipeline, se da un lato limita le possibilità del sistema, dall'altro rende il sistema più sicuro facilitando le funzioni di *watchdog* e di controllo dell'intero sistema.

3.3 Architettura software basata su diversi livelli di automazione

In questo paragrafo viene illustrata la proposta di architettura per veicoli intelligenti oggetto di questa tesi di dottorato. Visto lo stato dell'arte esposto nei paragrafi precedenti, unitamente alle richieste introdotte dal progetto OFAV e all'esperienza accumulata durante le DARPA Challenge e VIAC, si è deciso di creare un'architettura per veicoli intelligenti in grado di supportare diversi livelli di automazione che avesse le seguenti linee guida:

- **Modularità:** come visto in precedenza l'implementazione di un veicolo automatico è un'attività molto complessa, di conseguenza è necessario creare un'architettura che sia il più possibile modulare per facilitare l'attività di sviluppo da parte dei vari team di ricerca.
- **Scalabilità:** dovendo supportare più livelli di automazione deve essere semplice aggiungere funzionalità al sistema, per questo motivo l'architettura deve scalare linearmente al crescere dei bisogni di automazione del veicolo.
- **Fluidità:** un veicolo intelligente è soggetto a grandi cambiamenti durante il suo ciclo di sviluppo, risulta quindi necessario creare un'architettura che permetta di modificare e se necessario stravolgere intere sezioni software senza intaccare le altre.
- **Apertura:** l'architettura, come richiesta dal progetto OFAV, deve essere aperta, cioè deve poter essere implementata senza la necessità di utilizzare elementi con licenza proprietaria o di tipo "closed source". Questo riguarda in particolare le librerie utilizzate e alcuni dati sensibili come ad esempio le mappe cartografiche, che spesso sono chiuse ed a pagamento.
- **Fusione sensoriale multilivello:** l'attività di percezione, come visto nel paragrafo 1.1, è una delle attività più complesse per un veicolo intelligente. Per facilitare il compito dell'attività di percezione deve essere possibile effettuare una fusione sensoriale su più livelli, in modo da ottenere il massimo da ogni sensore disponibile.
- **Separazione tra i layer:** i tre classici layer di percezione, pianificazione e attuazione devono essere totalmente separati. In questo modo si ottiene una riduzione della ridondanza della percezione ed una maggiore semplicità e generalità nella programmazione di ogni elemento dell'architettura. Inoltre, in questo modo, ogni layer può essere testato indipendentemente dagli altri, facendo ad esempio uso di un simulatore.

- Approccio gerarchico alla pianificazione: la pianificazione in ambiente automotive è un'attività molto complessa, che comprende il prendere un gran numero di decisioni, soprattutto per tratte stradali che possono essere anche molto lunghe. Per questo motivo, e per permettere una pianificazione e soprattutto una ripianificazione entro i limiti imposti dall'esecuzione in "real time", si dovrà avere un approccio gerarchico alla pianificazione in modo da scomporre il problema generale in più sottoproblemi.
- Supporto a diversi livelli di autonomia: come visto nel capitolo 2 esistono diversi livelli di autonomia per un veicolo intelligente. L'architettura dovrà essere in grado di supportarli, unitamente alla cooperazione fra guidatore e veicolo. Da questo punto di vista la sfida consiste nell'integrare ottimamente i vari livelli in modo da minimizzare l'overhead di lavoro nella programmazione di ogni livello ed in modo da massimizzare il riuso di codice attraverso livelli diversi.

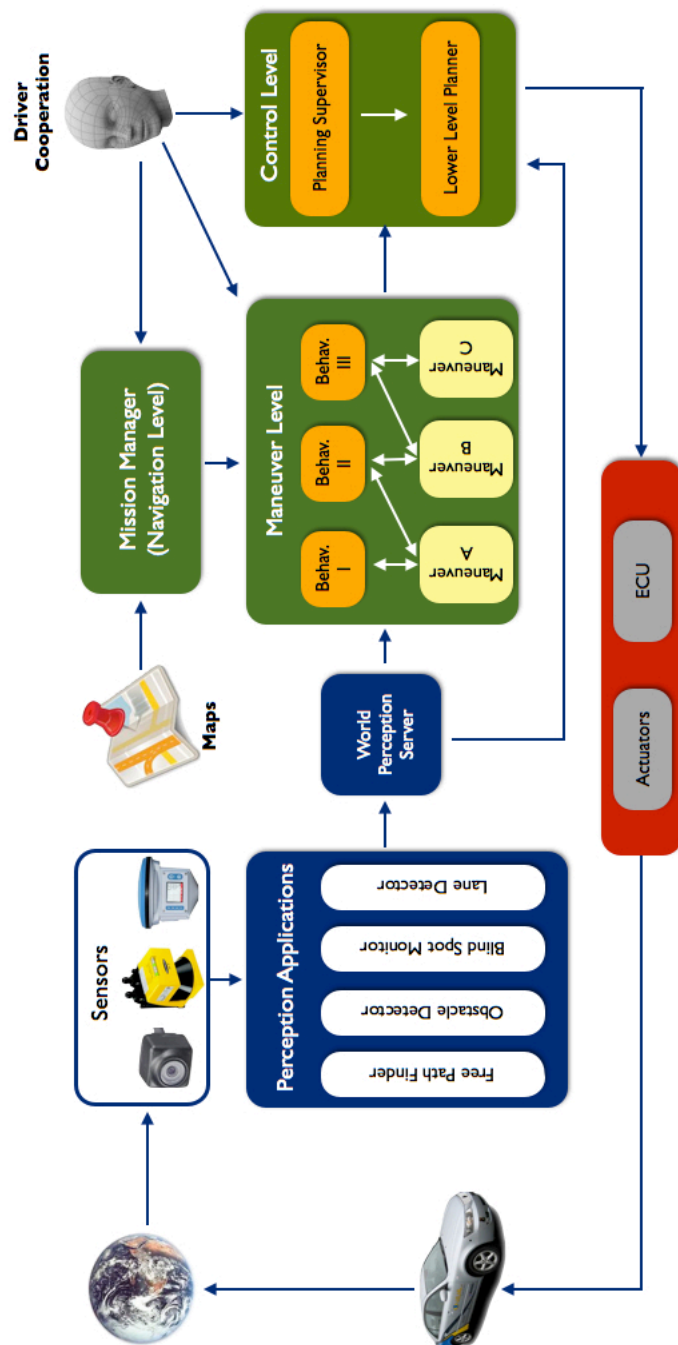


Figura 3.3: Architettura software basata su diversi livelli di automazione.

Il risultato della progettazione dell'architettura è raffigurato in figura 3.3. I tre layer evidenziati precedentemente sono raffigurati in colori diversi all'interno dell'immagine: il layer di percezione è raffigurato in blu, quello di pianificazione in verde e quello di attuazione in blu; i layer sono totalmente separati fra di loro, infatti non vi sono interdipendenze tra di essi. Molto importante è la separazione fra i layer di percezione e pianificazione che è stata ottenuta tramite l'introduzione del *World Perception Server*, che descriveremo a breve. Un fattore molto importante dell'architettura è lo schema di comunicazione: ogni elemento è un modulo indipendente che funge da servizio e riceve/trasmette dati tramite un protocollo di scambio messaggi appositamente sviluppato e chiamata *Vislab Bus* (VLBUS); il VLBUS è descritto nel prossimo paragrafo.

3.3.1 Vislab Bus

VLBUS scambia informazioni fra applicazioni e threads sulla stessa o su differenti macchine fisiche in una maniera completamente trasparente ed efficiente; in questo modo tutte le componenti del sistema possono essere realizzate come unità separate. VLBUS è una astrazione che modella la comunicazione inter-processo come un bus di dati: si possono avere diversi bus di dati, ciascuno dei quali con diverse linee indipendenti che rappresentano ognuna un diverso topic di comunicazione. Il modello utilizzato è quello del bus dati, al quale possono essere connessi contemporaneamente più dispositivi lettori/consumatori e più dispositivi scrittori/produttori.

Il mezzo fisico di comunicazione utilizzato è la *shared memory*, ovvero una porzione di memoria RAM mappata nello spazio di indirizzamento di tutti i processi; grazie a questo meccanismo, ai dati qui contenuti possano accedere processi diversi contemporaneamente. Questo meccanismo è molto rapido, diversi ordini di grandezza più veloce che accedere ad un file su disco.

In questo spazio di memoria condiviso vengono inserite dagli scrittori delle strutture dati, tramite le quali i diversi lettori hanno accesso alle informazioni. Le strutture utilizzate sono delle liste condivise - basate su *shared pointer* - ossia un meccanismo che permette di monitorare se e quanti lettori devono ancora consumare ciascun messaggio. Questo sistema consente di eliminare automaticamente un messaggio quando

è stato consumato dall'ultimo lettore o - nel caso che non vi siano attualmente lettori - di cancellarlo istantaneamente. Per evitare che più processi possano scrivere contemporaneamente nella stessa locazione di memoria è stato introdotto un meccanismo di sincronizzazione nella shared memory.

Anche i puntatori utilizzati sono differenti dai puntatori canonici: essendo la shared memory una porzione di memoria mappata nello spazio di indirizzamento di diversi processi, la stessa locazione di memoria viene vista con indirizzi differenti dai diversi processi. Per ovvi motivi non è quindi possibile utilizzare puntatori canonici. Si ricorre quindi ad un particolare tipo di puntatori detti offset pointer, che memorizzano l'indirizzo come differenza tra la propria posizione e la posizione del dato puntato. Non si tratta più di indirizzo assoluto ma di indirizzo relativo, ossia un dato che mantiene la propria consistenza anche tra processi differenti (con spazi di indirizzamento diversi).

Utilizzando la shared memory si possono quindi condividere le informazioni tra più processi sulla stessa macchina; per poter estendere questa capacità anche a processi su macchine diverse è stato realizzato un server collegato in lettura/scrittura al BUS. Questo server funge da *gateway* tra il BUS ed altre interfacce di comunicazione, permettendo la sincronizzazione di shared memory di computer diversi. Un esempio basato su reti locali consiste in un server che - tramite una interfaccia di rete - comunica con gli altri computer della stessa rete, e utilizzando il protocollo UDP/IP invia in multicast le informazioni che vengono inserite nella shared memory. Analogamente - quando riceve le informazioni generate dagli altri server le inserisce nella shared memory. Dei meccanismi analoghi sono stati sviluppati anche per TCP/IP e possono facilmente essere estesi ad altri protocolli di comunicazione, come ad esempio il CAN BUS.

Assieme a questo meccanismo di comunicazione sono forniti diversi strumenti che facilitano la trasformazione di dati arbitrari in dati inviabili su questo BUS. E' quindi possibile inviare dati grezzi, come sequenze alfanumeriche che dati codificati sotto forma di documenti XML; la scelta fra queste codifiche dipende dalla dimensione dei dati che si vuole inviare, dalla leggibilità che si vuole avere delle informazioni in transito, o dalla compatibilità necessaria con piattaforme diverse. Ad esempio se si

vogliono inviare delle informazioni dimensionalmente importanti (quindi lente da inviare) come delle immagini, conviene inviare i dati grezzi vincolandosi ad utilizzare delle macchine compatibili dal punto di vista dell'*endianness*.

Le principali componenti di ciascun livello sono analizzate nei prossimi paragrafi.

3.3.2 World Perception Server

Il World Perception Server ha il compito di raccogliere le informazioni dai moduli di percezione, unificarle e renderle disponibili al layer di pianificazione. Il World Perception Server ha due funzioni fondamentali:

- Separazione dei livelli: separa e rende indipendenti i layer di percezione e pianificazione, infatti ogni modulo di percezione deve solo occuparsi di scrivere dati sul WPS mentre ogni modulo di pianificazione deve solo occuparsi di ricevere dati dal WPS.
- Fusione Sensoriale: tramite il WPS è possibile effettuare una fusione sensoriale di alto livello, cioè una fusione di tipo track-to-track [43] [44]. Il metodo attraverso il quale la fusione viene effettuata è indipendente dall'architettura del WPS, come vedremo in seguito.

Lo schema generale del WPS e della sua struttura è riportato in figura 3.4. Il WPS possiede un'architettura interna a pipeline; ogni blocco riceve in input dei dati che elabora e rende poi disponibili ad altri blocchi di elaborazione, o alla parte di navigazione se il dato è considerato rifinito. Il tutto avviene in maniera completamente asincrona, in modo da massimizzare l'impegno delle unità di elaborazione e di velocizzare il più possibile il ritardo di ogni dato [45].

Ogni blocco di figura 3.4 rappresenta una funzionalità di fusione sensoriale completamente indipendente dalle altre; infatti grazie all'utilizzo del VLBUS come protocollo di scambio dati ogni modulo ha bisogno di conoscere solamente quante e quali linee di quanti e quali bus deve ricevere come input e quale accoppiata bus/linea deve utilizzare come output. Questo facilita e semplifica la scrittura di ogni blocco di fusione sensoriale, dando al contempo maggiore generalità al software scritto e massimizzandone quindi la possibilità di riuso nel tempo. L'output di ogni blocco può

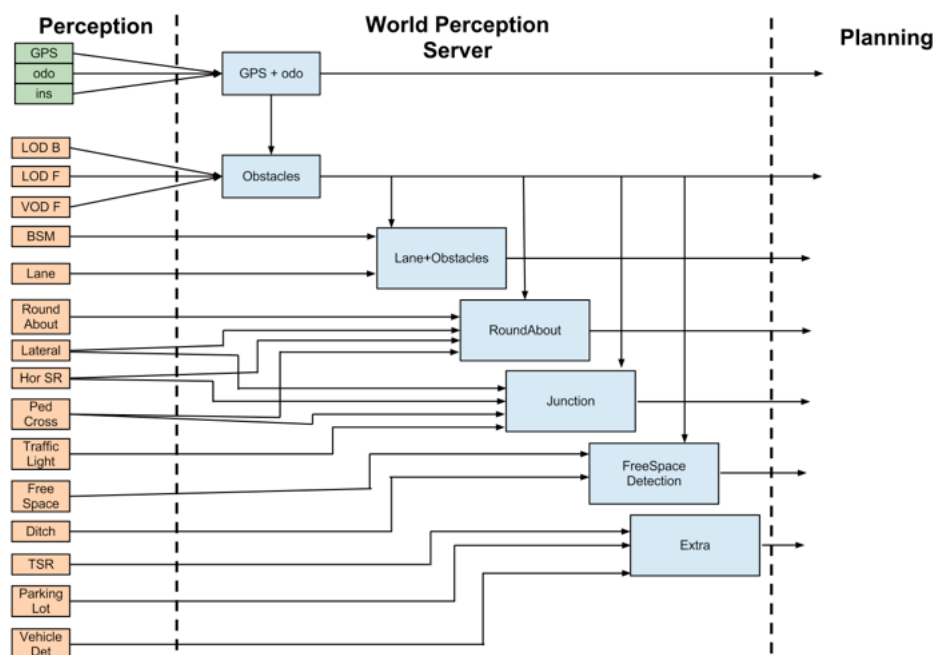


Figura 3.4: Schema a blocchi del WPS.

essere utilizzato anche da altri blocchi, in modo da creare una cascata di funzionalità semplici che implementano una funzionalità complessa. Ad esempio, sempre in figura 3.4 si ha che la funzionalità di fusione sensoriale che unisce i dati dei moduli di percezione Laser Obstacle Detector Front (MOD F), Laser Obstacle Detector Backward (MOD B) e Vision Obstacle Detector (VOD) è chiamata "Obstacles" e ha il compito di generare la mappa degli ostacoli sulla scena. Questo dato viene utilizzato poi sia da blocchi che ne rifiniscono ulteriormente il risultato, come il blocco LaneObstacle, sia da blocchi che hanno lo scopo di produrre risultati diversi, ma gli stessi dati vengono anche utilizzati "puri" dal pianificatore a livello di controllo per la generazione delle traiettorie.

L'utilizzo del VLBUS permette anche di avere una grande solidità interna a tutta l'architettura: se per un qualsiasi motivo una delle applicazioni di percezioni dovesse smettere di funzionare, per esempio per un errore hardware, l'intero WPS potrebbe continuare a funzionare utilizzando semplicemente le linee VLBUS che rimangono funzionanti e riducendo il grado di fusione ai soli sensori che ha disponibili. Ad esempio se il modulo detto LOD B dovesse smettere di funzionare il WPS continuerebbe comunque a ricevere dati da altre due applicazioni di percezione e l'intero sistema potrebbe continuare a funzionare. Allo stesso modo se uno dei moduli di fusione dovesse fallire i pianificatori dovrebbero semplicemente collegarsi ai canali dei moduli di fusione dei dati più grezzi per ricevere dati dello stesso tipo, ma meno rifiniti. In

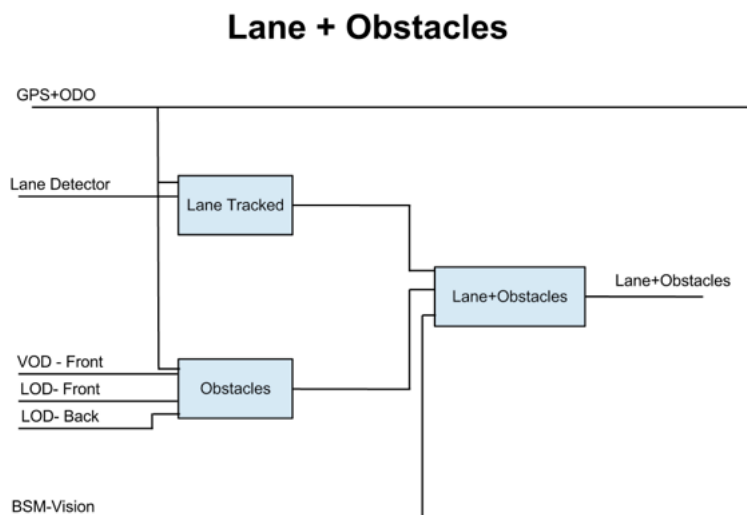


Figura 3.5: Composizione del blocco di fusione sensoria Lane+Obstacles.

generale le applicazioni di fusione sensoriale vengono divise in due grandi famiglie:

- Fusione a basso livello: le applicazioni che appartengono a questa famiglia usano i dati puri di ciascun sensore e li uniscono per ricavarne un dato miglio-

re di quello che avrebbero ottenuto analizzandoli separatamente. Questo tipo di fusione trae vantaggio dalla grande quantità di dati che si possono raccogliere a basso livello ed ha il difetto che non è generalizzabile, in quanto un algoritmo scritto per una certa situazione con una certa configurazione di sensori difficilmente funzionerà con un'altra configurazione, o con altri sensori. Questo avviene poiché per rendere efficace questo tipo di fusione si devono sfruttare in modo approfondito le caratteristiche intrinseche di ciascun sensore, caratteristiche che non sono ritrovabili in sensori diversi o addirittura in altri modelli dello stesso tipo di sensore. Un'esempio di questo tipo di fusione verrà presentato nel capitolo.

- Fusione ad alto livello: detta anche "track-to-track", le applicazioni appartenenti a questa famiglia si occupano di fondere dati già elaborati; appartengono a questa famiglia tutte le applicazioni che uniscono gli ostacoli trovati con tecnologie diverse in modo da creare un'unica, omogenea, mappa degli ostacoli. La fusione ad alto livello ha il grande vantaggio di essere generalizzabile e modulare, per cui un algoritmo scritto per effettuare un fusione track-to-track di ostacoli funzionerà sempre qualsiasi sia la tecnologia che fornisce le varie liste di ostacoli da fondere. Per contro, la fusione di alto livello sfrutta meno le peculiarità di ciascun sensore, ma trae vantaggio principalmente dalle diverse zone di funzionamento dei sensori stessi.

I moduli del WPS appartengono principalmente alla famiglia della fusione ad alto livello, in quanto operano principalmente su dati già elaborati. Ne è un esempio il blocco chiamato "Obstacles" che prende in ingresso tre liste di ostacoli e cerca di fonderle utilizzando i dati odometrici del veicolo. Unica eccezione risulta essere il blocco "GPS+IMU" di figura 3.4 che utilizza dati puri provenienti dal sensore GPS e inerziale e di conseguenza effettua una fusione di basso livello. Un'altro esempio di fusione a basso livello nell'architettura verrà presentata nel capitolo 4.

I tipi di dato in uscita dal WPS ed utilizzabili quindi a livello di pianificazione sono i seguenti:

- Ostacoli: lista degli ostacoli sulla scena, proveniente dalla fusione dei dati di diversi moduli di percezione ostacoli.
- Corsie: lista e posizione delle corsie, pesate nel tempo e fuse con i dati odometrici.
- Ostacoli + Corsie: lista delle corsie, con inseriti gli ostacoli presenti sulla corsia stessa. Da sottolineare che il blocco in figura 3.4 è una semplificazione del blocco reale che è invece raffigurato in figura 3.5, in cui è possibile notare che l'informazione aggregata comprende anche la visione del "Blind Spot Monitor" per determinare se la corsia sinistra sia occupata o meno.
- Rotonde ed incroci: queste due informazioni vengono ricavate dalla fusione sensoriale di diversi moduli di percezione, tra cui il rilevatore di corsie, semafori, rotonde e linee stradali.
- Spazio libero: questo dato rappresenta la posizione dello spazio libero di fronte al veicolo, dove per spazio libero non si intende libero da ostacoli ma bensì si vuole identificare la posizione del tracciato stradale di fronte al veicolo.
- Altri: le altre informazioni sensoriali quali cartelli stradali o parcheggi sono fuse in un unico canale riservato a questi eventi "sporadici".

3.3.3 Layer di Pianificazione

La pianificazione in ambiente automotive è un compito molto complesso che, come già visto in precedenza, richiede una gran numero di decisioni nel corso di un viaggio, soprattutto se il viaggio è di lunga tratta. Risulta quindi indispensabile dividere l'attività di pianificazione su 3 diversi layer [46], detti layer di Navigazione, Manovra e Controllo, come già trattato nel capitolo 2. I tre layer sono raffigurati in verde in figura 3.3; la comunicazione fra i vari moduli intra ed extra layer avviene sfruttando il protocollo del VLBUS. L'approccio gerarchico alla pianificazione è mostrato in figura 3.6; ricevendo come input un punto di destinazione il pianificatore a livello di navigazione produce un percorso, composto da una serie di strade ed incroci, che viene

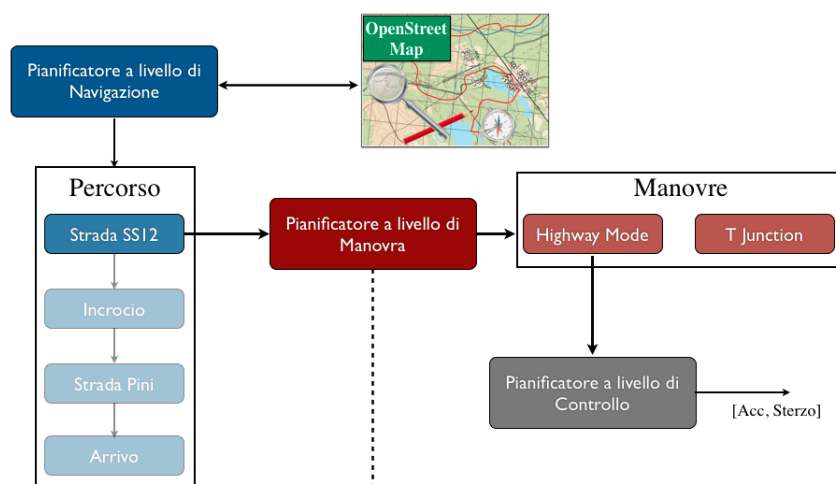


Figura 3.6: Approccio gerarchico alla pianificazione, un esempio.

passato al pianificatore a livello di manovra. Questo pianificatore, per ogni elemento della lista, calcola uno od un set dei behaviour visti nel capitolo 2 che permetteranno al veicolo di percorrere l'elemento desiderato; questo set viene infine utilizzato dal pianificatore a livello di controllo per generare una traiettoria percorribile dall'utente, traiettoria che viene tradotta in azioni dal layer di attuazione. In questa tesi di dottorato mi concentro principalmente sui livelli di navigazione e manovra poiché sono i livelli in cui ho dato il contributo maggiore.

Il primo layer è quello di navigazione: il suo scopo è quello di generare un percorso da percorrere per arrivare alla destinazione indicata dall'utente. Per scopi di navigazione vengono quindi utilizzate delle mappe GPS ma esse vengono utilizzate solamente come fonte cartografica: il loro scopo infatti è solamente quello di permettere al pianificatore a livello di navigazione di decidere quali strade percorrere, a livello cartografico, per giungere a destinazione.

Le mappe utilizzate in questo processo, in rispetto al vincolo esplicitato in precedenza di gratuità ed apertura dei dati utilizzati, provengono dal progetto "Open Street Map" [47]. Questo progetto ha lo scopo di creare una mappa gratuita, libera

ed editabile dell'intero mondo. Le mappe estratte dai server del progetto OSM sono codificate in XML e contengono un gran numero di layer, ognuno dei quali è dedicato ad un tipo di informazione diversa. Per gli scopi di questa architettura servono una quantità molto minore di informazioni, di conseguenza le mappe scaricate vengono filtrate in modo da mantenere solamente i dati necessari alla navigazione di un veicolo intelligente. Dopo questo passaggio si ha una mappa contenente una serie di punti,

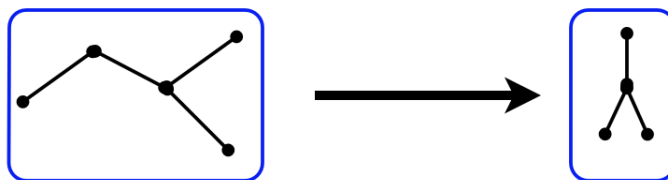


Figura 3.7: Road Junction Map: compressione degli archi.

detti nodi, ed una serie di gruppi, le strade, che identificano i nodi collegati fra di loro, aggiungendo una serie di proprietà. Per diminuire il numero di nodi delle mappe, diminuendo di conseguenza la dimensione della mappa in memoria e allo stesso tempo velocizzando la velocità di pianificazione, si è deciso di apportare le seguenti modifiche alla mappa:

- Road Junction Map: la mappa è stata modificata e compressa in un nuovo concetto di mappa, chiamata appunto Road Junction Map, memorizzata in un grafo. Come si può vedere in figura 3.7 nel nuovo grafo ogni nodo rappresenta un bivio fra due o più strade mentre ogni arco rappresenta una strada. Questo tipo di mappa, come vedremo poi, presenta grandi vantaggi dal punto di vista del numero di nodi;
- Compressione: il grado caricato in memoria è stato ulteriormente compresso e salvato in memoria in formato *raw*, senza tornare alla memorizzazione XML di OSM. Questo permette di ridurre notevolmente i tempi di caricamento della mappa all'avvio del sistema.

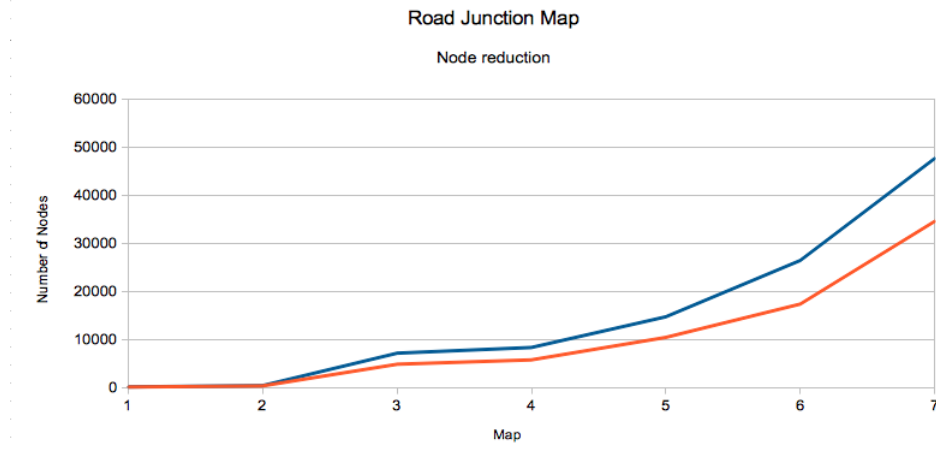


Figura 3.8: Road Junction Map: risultati.

I risultati delle operazioni fatte sulle mappe sono riportati nel grafico di figura 3.8, in cui la linea blu rappresenta i nodi della mappa OSM e la linea arancio rappresenta il numero di nodi della Road Junction Map. I nodi della Road Junction Map sono di quasi il 25% minori dei nodi della mappa OSM in una mappa che, per numero di nodi (50k), è grande all'incirca come quella dell'intera regione Emilia-Romagna; il risparmio in termini di nodi si riflette in una maggiore velocità di caricamento della mappa stessa, ma soprattutto in una maggiore velocità di pianificazione.

Il pianificatore utilizzato a livello di navigazione è un pianificatore di tipo euristico, basato in particolare su A*. Si è scelto un pianificatore di questo tipo in quanto, con una euristica ben definita, garantisce il percorso ottimo con un tempo di calcolo generalmente piuttosto limitato, così come già evidenziato in [48] [49]. Nei pianificatori euristici, ed in A* in particolare, vi sono due componenti importanti, più precisamente la funzione euristica e la funzione di costo.

$$c(x) = d(x, x-11) * v(x) + \delta(x) \quad (3.1)$$

$$h(x) = D(x, x_f) * k_v \quad (3.2)$$

La funzione di costo, descritta dall'equazione 3.1, è costituita dal tempo di percorrenza dell'arco, calcolato basandosi su lunghezza e velocità possibile nell'arco stesso a cui viene sommato un ritardo costante $\delta(x)$; la funzione euristica è invece costituita dal tempo di percorrenza dalla posizione attuale al goal calcolando la distanza in linea retta tramite le coordinate GPS e supponendo una velocità media costante di 50 km/h.

I vantaggi di questo approccio sono i seguenti:

- Percorsi veloci: l'utilizzo del tempo di percorrenza e non della distanza pura permette di indirizzare il pianificatore verso percorsi su strade ad alto scorrimento, piuttosto che su strade di città;
- Percorsi non tortuosi: la costante $\delta(x)$ sommata ad ogni nodo fa sì che il pianificatore preferisca rimanere sulla stessa strada piuttosto che cambiarla per una minimamente più veloce. Questo evita il crearsi di percorsi tortuosi e mantiene il veicolo sulla stessa strada più a lungo.

I risultati del pianificatore implementato sono riportati in figura 3.9; come si può notare il tempo di pianificazione rimane su tempi ragionevoli anche per mappe di grande dimensioni, rendendo anche il replanning fattibile in real-time, se necessario.

Come detto nell'introduzione di questo capitolo questa tesi di dottorato, per quanto riguarda il layer di pianificazione, tratta principalmente i livelli di navigazione e manovra. Il livello di manovra in particolare è stato esposto nel capitolo 2, quindi per una sua descrizione più approfondita si rimanda a quel capitolo. Un fattore molto importante nella progettazione del layer di pianificazione è stata la volontà di ottenere una perfetta integrazione fra tutti i livelli di automazione evidenziati nel capitolo 1. Di conseguenza, le componenti software utilizzate durante una sessione di digital driving sono esattamente le stesse utilizzate durante una sessione completamente automatica. In generale, tutte le componenti di un livello vengono utilizzate anche nel

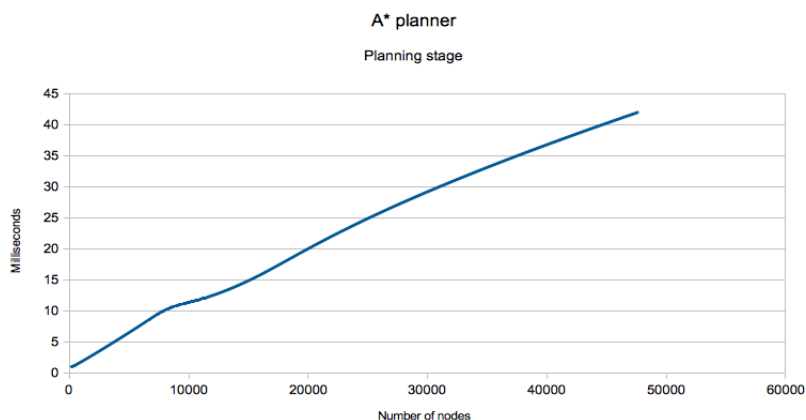


Figura 3.9: A* planning: risultati.

livello soprastante, semplicemente nel nuovo livello verrà aggiunta una componente software (o verranno aggiunte funzionalità ad una componente esistente) che si occuperà di tutte le operazioni che nel livello sottostante erano lasciate al conducente. In particolare, il modulo intelligente detto "Mission Manager" in figura 3.3 è stato progettato per svolgere mansioni diverse in ogni livello: in caso di sessione completamente automatica il Mission manager viene utilizzato quale pianificatore di alto livello, con lo scopo di definire il percorso dalla partenza alla destinazione e con lo scopo di attivare al giusto momento le manovre necessarie a compiere l'intero tragitto. Viceversa, in caso di sessione di guida con Digital Driving il Mission Manager diviene un assistente intelligente alla guida, con lo scopo di comunicare al guidatore, attraverso le apposite HMI, quali manovre sono fattibili in ogni momento.

3.3.4 Layer di Percezione

Il layer di percezione ha lo scopo di raccogliere le informazioni sensoriali, elaborarle ed infine inviarle al WPS; la comunicazione intra ed extra layer avviene attraverso il VLBUS, mentre la connessione con i sensori è gestita attraverso l'interfaccia sensoriale del framework *Generic Obstacle and Lane Detector* (GOLD) del

VisLab [50] [51]. In particolare ogni modulo di percezione è strutturato come un servizio indipendente, con una lista di dati sensoriali in ingresso e una o più accoppiate bus/linea del VLBUS come uscita. L'ontologia del canale di uscita è condivisa all'interno dell'intera architettura, in modo da rendere consistente la descrizione del mondo all'interno del WPS. In questo layer è necessario avere una forte modularità e fluidità dell'architettura, poiché i moduli di percezione sono molteplici e il loro sviluppo causa forti e veloci cambiamenti; un elenco dei principali moduli di percezione all'interno del progetto OFAV è visibile in figura 3.4.

Durante i tre anni di dottorato ho sviluppato diversi moduli di percezione, una descrizione dei quali è riportata nel prossimo capitolo.

Capitolo 4

Moduli di percezione

*In questo mondo non vediamo le cose come sono.
Le vediamo come siamo, perché ciò che vediamo
dipende principalmente da ciò che stiamo cercando.*

– John Lubbock

4.1 Rilevamento della superficie stradale basata sul colore

Il rilevamento della superficie stradale è una delle attività chiave nel campo dei sistemi di aiuto alla guida; in particolare questa attività riguarda il trovare la posizione e dimensione della superficie percorribile davanti al veicolo senza fare affidamento a tecnologie di visione tridimensionale ma basandosi solamente su informazioni derivanti dallo studio del colore delle immagini acquisite. Questo modulo si configura come modulo base d'aiuto per gli altri moduli, è quindi fondamentale che abbia tempi di esecuzione molto brevi, per poter essere eseguito sempre in parallelo anche con altri moduli, che possa funzionare in tutte le condizioni stradali e che presenti poche precondizioni.

In letteratura, come descritto in [52], sono stati proposti vari metodi per identificare la strada basati sulla visione, e il colore è sempre stato usato come informazione

dominante. Aggiungendo altre informazioni come la texture, la struttura della strada e la profondità, si possono migliorare le performance dei diversi algoritmi, ma allo stesso tempo si va a ridurre la generalità del metodo. Lo scopo del modulo che si vuole sviluppare è ottenere un *path detector*, quindi un identificatore non solo dell'asfalto, ma di qualsiasi tipo di terreno che si sta percorrendo (sia esso asfalto o sterrato). Per far questo, non possiamo accettare limitazioni sulla forma della strada o sull'orizzonte. In sostanza, si vuole implementare un *asphalt detector* che funzioni su tutte le superfici. Risulta quindi necessaria l'identificazione dello spazio di colore opportuno per individuare la carreggiata e un algoritmo di apprendimento che possa essere robusto in tutte le situazioni in analisi.

L'approccio che si è utilizzato è quello di apprendere, rispetto a ciò che abbiamo davanti, qual è la giusta via da seguire. Per far questo serve qualcosa che ne descriva univocamente l'aspetto: un descrittore. Un descrittore identifica in maniera univoca un certo elemento, il primo passo nello sviluppo dell'algoritmo consiste quindi nel trovare una caratteristica peculiare della carreggiata, che possa essere utilizzata come elemento discriminante all'interno dell'algoritmo di riconoscimento. Il descrittore più semplice è definito dal colore e di conseguenza si sono analizzati diversi spazi di colore: CIELab, HSV, HunterLab e RGB. Di seguito riporto solo le considerazioni più importanti su CIELab e HSV.

4.1.1 Analisi degli spazi di colore CIELab e HSV

Lo spazio di colori di tipo Lab è basato sulla teoria dei colori opposti, che assume che i recettori nell'occhio umano percepiscono il colore come coppie di opposti:

- Luce – buio.
- Rosso – verde.
- Giallo – blu.

In Lab, L esprime la luminanza, e a e b le dimensioni del *color-opponent* basate su coordinate non lineari. A differenza dei modelli di colore come RGB o CMYK, Lab approssima la visione umana.

Dato che lo spazio di colore Lab è più vasto della gamma di colori nei display dei computer e della visione umana, un'immagine bitmap rappresentata in Lab richiede più dati per pixel per ottenere la stessa precisione dell'immagine RGB o CYMK. CIEL*a*b* è lo spazio di colori più completo in quanto descrive tutti i colori visibili dall'occhio umano. Le tre coordinate di CIEL*a*b*, il cui andamento è presentato in

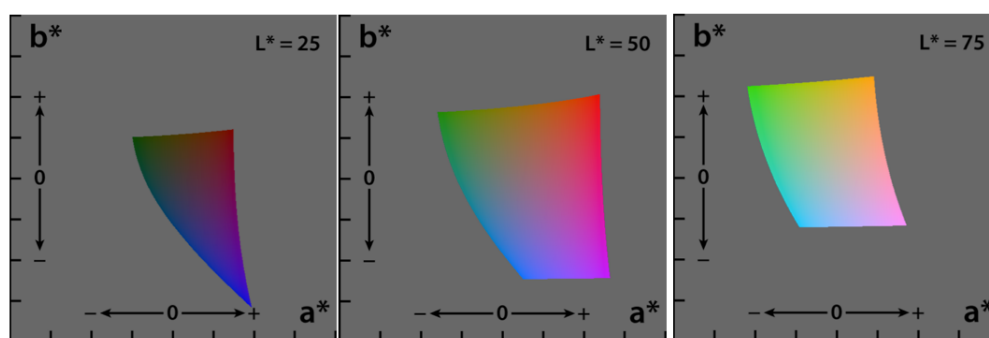


Figura 4.1: Spazio di colore CIELab: rapporto a/b al variare di L.

figura 4.1 rappresentano:

- la luminosità del colore ($L^* = 0$ indica il nero e $L^* = 100$ rappresenta il bianco diffuso);
- la posizione tra rosso/magenta e verde (a^* , un valore negativo indica il verde, mentre un valore positivo indica il magenta);
- la posizione tra giallo e blu (b^* , un valore negativo indica il blu, mentre un valore positivo indica il giallo).

Le relazioni non lineari per L^* , a^* e b^* simulano la risposta non lineare dell'occhio. Non ci sono formule semplici e dirette per effettuare una conversione da RGB a CIEL*a*b* poiché RGB è un modello di colore dipendente dal dispositivo, quindi deve essere trasformato in uno spazio di colori assoluto per essere poi portato nello spazio di colori CIEL*a*b*.

In questa applicazione si è fatta prima una conversione dallo spazio di colori RGB a quello XYZ e poi da XYZ a CIEL*a*b*, questo perché lo spazio di colori RGB non sempre riesce a produrre un colore equivalente ad ogni lunghezza d'onda mentre XYZ è stato creato in modo da produrre ogni colore con valori di tristimolo positivi. Lo spazio di colori XYZ è uno dei primi modelli definiti matematicamente, derivante da una serie di esperimenti sostenuti verso la fine degli anni '20. Risulta essere uno dei modelli più usati e spesso serve come base per definire altri spazi di colore. I valori di X, Y e Z sono parametri ricavati dal colore rosso, verde e blu, e sono legati allo spazio di colori RGB da una trasformazione lineare. Il valore della Y descrive la luminanza del colore: il fatto che questa possa essere costruita attraverso una combinazione lineare di colori RGB non è garantito, ma ci si può aspettare che sia vero a causa della natura quasi lineare della vista umana. La conversione adottata è quindi la seguente:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.41240 & 0.35760 & 0.1805 \\ 0.21260 & 0.71520 & 0.0722 \\ 0.01930 & 0.11920 & 0.9505 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (4.1)$$

$$L^* = 116f(Y/Y^n) - 16 \quad (4.2)$$

$$a^* = 500[f(X/X^n) - f(Y/Y^n)] \quad (4.3)$$

$$b^* = 200[f(Y/Y^n) - f(Z/Z^n)] \quad (4.4)$$

$$f(t) = \begin{cases} t^{1/3}, & \text{se } t > (6/29)^3 \\ 1/3(29/6)^2 t + (4/29), & \text{altrimenti} \end{cases}$$

Un altro spazio di colore analizzato è quello definito Hue(tonalità), Saturation(saturazione) and Value(valore) (HSV). Per saturazione si intende l'intensità e la purezza del colore, la luminosità (il valore) è un'indicazione della sua brillantezza e infine la tonalità

indica il colore stesso. HSV fornisce una delle rappresentazioni più comuni di un modello di colori RGB in un sistema di coordinate cilindrico (come mostrato in figura 4.2), che riarrangia la geometria di RGB per far sì che questa sia più intuitiva e percettualmente rilevante rispetto alla rappresentazione cartesiana, ed è definito come un cono distorto all'interno del cilindro. La tonalità H viene misurata dall'angolo intorno all'asse verticale in cui il rosso è a 0 gradi, il verde è a 120 e il blu a 240. La saturazione S va da zero (sull'asse del cono) ad 1 (sulla superficie del cono). Infine il valore (la luminosità) V rappresenta l'altezza del cono, con lo zero che indica nero e l'uno il bianco. Questo modello è stato sviluppato principalmente per applicazioni

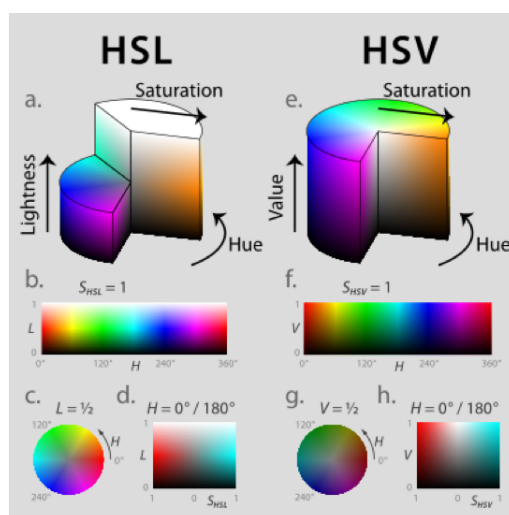


Figura 4.2: Sfere rappresentanti gli spazi di colore HSV e HSL.

di computer graphics, ed è utilizzato nei software di editing grafico per modificare il colore all'interno delle immagini e meno comunemente viene utilizzato per l'analisi di immagini e nella computer vision (ad esempio per eseguire *feature detection* o segmentazione di un'immagine).

4.1.2 Algoritmo per il riconoscimento della superficie stradale

L'algoritmo sviluppato per il riconoscimento della superficie stradale è raffigurato in figura 4.3. L'algoritmo è costituito dai seguenti passi:



Figura 4.3: Diagramma rappresentante l'algoritmo di rilevazione della superficie stradale.

- Applicazione del filtro di media: si esegue su tutta l'immagine, un filtraggio di media su un intorno per sfocare la scena e migliorare l'output.
- Zona di apprendimento: viene definita la zona sulla quale determinare i valori validi per la superficie stradale nella fase di apprendimento. Durante il primo apprendimento si ipotizza che la zona di acquisizione sia pulita, per gli apprendimenti successivi viene fatto un controllo sul pixel per evitare che vengano considerati come validi pixel non coerenti con il descrittore della superficie.
- Conversione del vettore dell'immagine nello spazio colore di destinazione: si effettua la conversione dell'immagine nello spazio colore desiderato: CIELab, HunterLab, HSV o RGB.

- Creazione del descrittore e Classificazione dei pixel dell'immagine: per quanto riguarda l'analisi dell'immagine e la definizione dei valori per i pixel considerati come appartenenti alla superficie stradale viene inizialmente analizzato ogni singolo pixel della finestra di apprendimento, calcolata la media e la deviazione standard totale dei valori dei pixel sull'intera finestra di apprendimento durante la fase di creazione del descrittore ed infine viene definito un pixel appartenente alla superficie stradale se soddisfa la disequazione 4.5.

$$\bar{x} - \alpha\sigma \leq x \leq \bar{x} + \alpha\sigma \quad (4.5)$$

4.1.3 Risultati

Un primo risultato a cui si è giunti nello sviluppo di questo modulo di percezione è quello relativo alla determinazione di quale sia il miglior spazio di colore da utilizzare per creare il descrittore. In figura 4.4 sono presentati i risultati riguardanti la classifi-

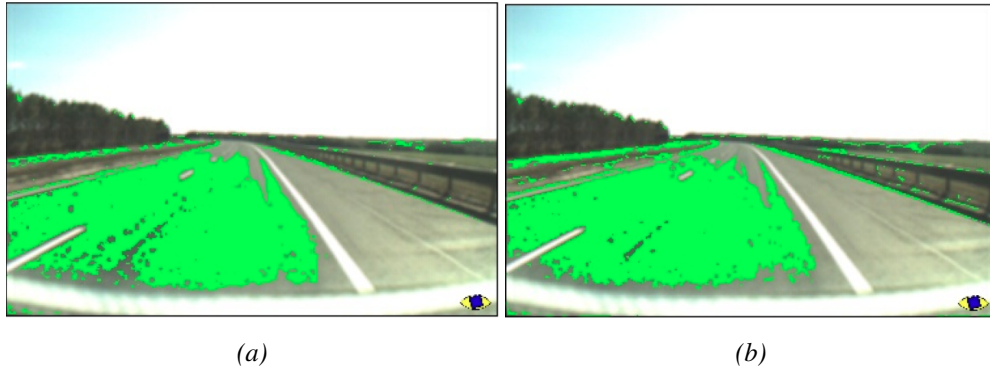


Figura 4.4: (a) Risultati con conversione CIE Lab (b) Risultati con conversione HSV.

cazione di una superficie di tipo autostradale con conversione HSV e con conversione CIE Lab. Com'è possibile vedere dalla figura, i risultati sono simili per entrambi i tipi di conversione, di conseguenza, essendo HSV più veloce nella fase di conversione rispetto a CIE Lab, si è scelto questo spazio di colore per la costruzione del descrittore della superficie stradale. Un ulteriore risultato è rappresentato ovviamente dal funzionamento dell'algoritmo nel riconoscimento di diverse superfici stradali, come

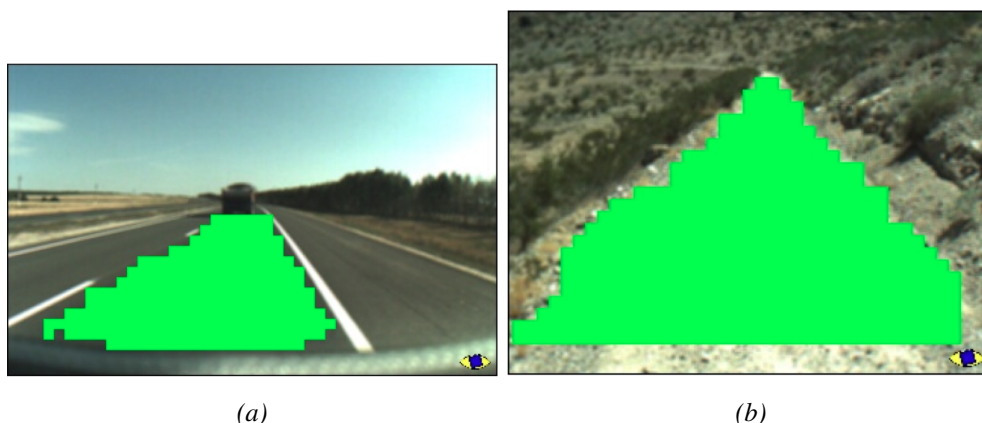


Figura 4.5: (a) Risultato su strada urbana (b) Risultato su sterrato.

raffigurato in figura 4.5, dove alla fine dell'algoritmo è stata applicata un'operazione di *tassellatura* per velocizzare l'algoritmo pur mantenendo una precisione sufficiente allo scopo di questo modulo di percezione. In entrambi i casi presentati l'algoritmo dimostra di avere un buon funzionamento e soprattutto dimostra di essere molto versatile al cambiare della superficie da riconoscere. Una possibile problematica è rappresentata dalla presenza di zone d'ombra all'interno dell'area di rilevazione, che, probabilmente, potrebbero causare una perdita di precisione dell'intero algoritmo che solo in parte può essere mitigata attraverso operazioni come la tassellatura. Da sottolineare che l'intero algoritmo, in esecuzione su un Intel Pentium Core 2 Duo 2.4GHz, occupa non più di 17 ms per ogni ciclo di esecuzione; di conseguenza, visti i risultati e le basse richieste computazionali, questo modulo può essere utilizzato come modulo base per la rilevazione della superficie stradale di fronte al veicolo intelligente.

4.2 Rilevamento e tracking di un veicolo Leader

Questo modulo di percezione ha lo scopo di individuare, tramite input di un utente, un veicolo che precede il veicolo intelligente e di inseguire il veicolo stesso nel tempo, individuandone la posizione rispetto al veicolo intelligente. Questo modulo risulta utile per applicazioni del paradigma Leader-Follower, o per implementazione di

tecniche di *platooning* automatiche, cioè per la creazione di convogli di veicoli intelligenti che seguono un veicolo leader [53]. Il paradigma del Leader-Follower risulta particolarmente utile in quelle situazioni di traffico (come ad esempio le situazioni di coda) dove la vicinanza tra i veicoli e lo stato del traffico impediscono il normale rispetto delle regole stradali, per cui il veicolo intelligente, se indirizzato dal guidatore, può essere programmato in modo da seguire un veicolo leader che lo precede.

In letteratura esistono diversi approcci per la ricerca ed inseguimento di un veicolo leader, e sono basati principalmente sull'utilizzo di telecamere; in particolare si sono distinti approcci che utilizzano una telecamera monoculare [54] [55] oppure approcci che utilizzano una coppia di telecamere stereoscopiche [56].

L'approccio utilizzato in questo modulo è leggermente diverso: si intende usare una telecamera monoculare per la rilevazione e l'inseguimento del veicolo in coordinate immagine, fondendo poi i dati ottenuti con quelli ricavati dalla segmentazione di un sensore Lidar frontale, in modo da ottenere una misura della distanza precisa ed affidabile. Per risolvere i problemi evidenziati in [54] [55] si è deciso di ricorrere ad uno spazio di colori invariante rispetto all'illuminazione, che verrà presentato nel prossimo paragrafo.

4.2.1 Immagini Intrinseche

L'elaborazione di immagini, soprattutto in ambiente automotive, è affetta comunemente da problemi dovuti all'illuminazione: immagini con forti ombre o forti saturature causano problemi agli algoritmi di elaborazione. Le immagini intrinseche sono particolari tipi di immagini libere da tutti gli effetti dell'illuminazione [57], grazie alla separazione, nell'analisi colorimetrica di un oggetto, fra la componente dovuta alla cromaticità e la componente dovuta all'illuminazione. Questo può avvenire sotto alcune condizioni ipotizzate: che la luce ambientale sia di tipo planckiano, che le superfici presenti nell'immagine riflettano la luce in modo lambertiano e infine che il sensore utilizzato abbia una risposta di tipo *narrow-band*. Queste assunzioni sono dette assunzioni PLN:

- Luce Planckiana: la luce ambientale naturale può essere considerata approssi-

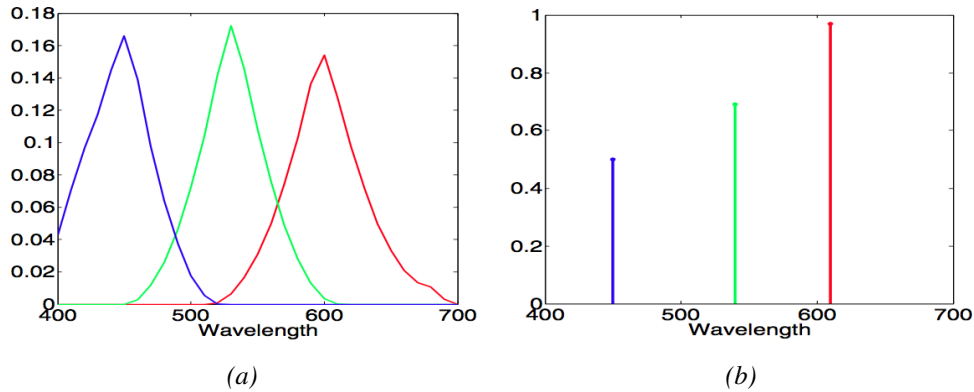


Figura 4.6: (a) Risposta in frequenza di una telecamere Sony DXC930 (b) Risposta in frequenza di un sensore narrow-band ideale.

mativamente una radiazione planckiana. Infatti per le lunghezze d'onda all'interno dello spettro del visibile, la radiazione giace su una specifica curva dello spazio dei colori CIE detta blackbody locus. La proprietà principale di questo tipo di illuminazione consiste nell'essere diffusa uniformemente da tutte le direzioni, senza evidenziare nessun punto preciso di provenienza della luce stessa;

- Superficie Lambertiana: questo tipo di superficie rappresenta una superficie ideale che riflette la radiazione incidente in modo uguale in tutte le direzioni. Un raggio di luce che colpisce una superficie lambertiana si riflette seguendo la legge di Lambert sulla riflessione diffusa: $I(\sigma) = I_0 \cos(\theta)$;
- Sensori a banda stretta (narrow band): le risposte dei sensori della telecamera sono approssimabili a delle δ di Dirac. In corrispondenza delle tre frequenze principali caratteristiche del sensore RGB, la risposta presenta un picco di sensibilità, mentre risulta molto meno sensibile verso le altre frequenze, com'è possibile notare in figura 4.6.

In realtà nessuna di queste condizioni è verificata nelle situazioni di utilizzo reale ma è stato provato in [58] che le approssimazioni sul sensore possono essere considerate

tollerabili per qualsiasi telecamera presa in analisi dagli autori, mentre le rimanenti approssimazioni effettuate sulla componente ambientale (luce approssima approssimativamente lambertiane) non portano a errori che compromettono il funzionamento dell'algoritmo. Sotto le assunzioni PLN è possibile convertire una data immagine

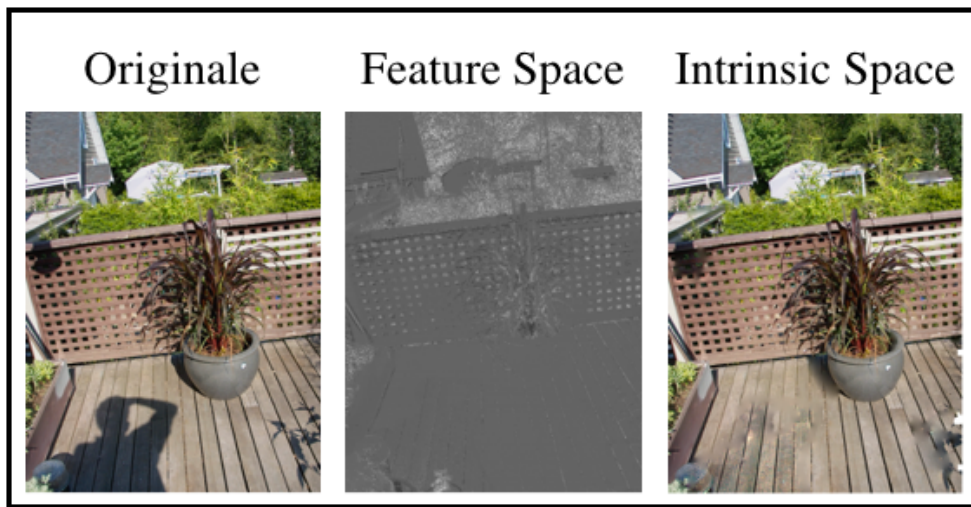


Figura 4.7: Intrinsic Images: processo di calcolo in due fasi.

RGB in una immagine libera da ombre I_{sf} attraverso una trasformazione in due fasi: in primo luogo, si ottiene una immagine in scala di grigio I , appartenente ad un feature space libero da ombre, partendo da I_{RGB} ; successivamente si ricava I_{sf} attraverso un processo di ricombinazione di I_{RGB} con I , come chiarito nell'immagine 4.7. Nella pratica, ottenere I è una procedura a livello di singolo pixel, e quindi può essere svolta in real-time; al contrario il calcolo di I_{sf} è una procedura di complessità a livello dell'intera immagine, e di conseguenza richiede molto più tempo di computazione. Per questo motivo l'applicazione di individuazione del veicolo sarà eseguita nel feature-space di I , e non nello spazio delle immagini intrinseche vere e proprie.

La trasformazione che porta dallo spazio di colori RGB allo feature space di I si basa sulla ricerca dell'angolo θ di invarianza rispetto all'illuminazione. Se R , G e B

sono i tre canali standard, si ponga :

$$\begin{cases} r = \log(R/G) \\ b = \log(B/G) \end{cases}$$

come i corrispondenti valori di *log-cromacity* usando il canale verde come canale di normalizzazione. Nello spazio di colori bidimensionale rb allora è possibile visualizzare sul piano le coordinate cromatiche in codifica logaritmica di ogni pixel (grafico LCD o *log-chromacity differences*). Allora, sotto le ipotesi PLN esposte precedentemente, si può notare che, variando le condizioni di illuminazione, superfici dello stesso colore vengono graficate su punti allineati a formare un fascio di rette parallele, il cui offset rappresenta la differenza di componente cromatica principale intrinseca della superficie. Inoltre, queste linee definiscono un asse ortogonale l_θ sul quale una stessa superficie colorata, sotto diverse condizioni di illuminazione, viene proiettata sullo stesso punto e a movimenti lungo l'asse l_θ corrispondono cambiamenti della componente cromatica principale della superficie, come mostrato in figura 4.8. In al-

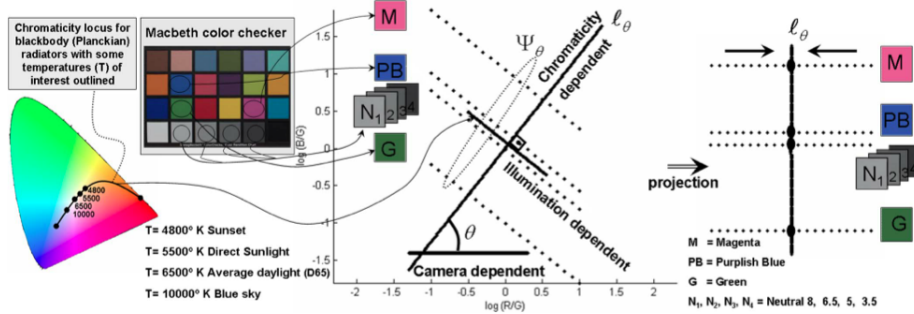


Figura 4.8: Intrinsic Images: meccanismo di funzionamento e asse l_θ .

tre parole l_θ è un asse monodimensionale che rappresenta la componente cromatica principale, la quale può essere rappresentata in scala di grigi in modo che ogni livello di grigio rappresenti una diversa componente in RGB; concludendo attraverso questa trasformazione si ottiene un'immagine in uno feature-space invariante rispetto alle condizioni di illuminazione.

Risulta quindi chiaro che ottenere θ è cruciale per ottenere il feature-space di I . Questo angolo è una funzione del sensore della telecamera e non è correlato alle condizioni di illuminazione o alla superficie di riflessione della luce; esso può quindi essere considerato come un parametro intrinseco di ogni telecamera e di conseguenza può essere calcolato attraverso una opportuna calibrazione off-line. In letteratura esistono diversi metodi per la calibrazione per immagini intrinseche [59] [60] [61]. Per questo modulo di percezione di è scelto di calibrare la telecamera utilizzando come criterio discriminante la minimizzazione dell'entropia, come in [59], ma con alcune modifiche.

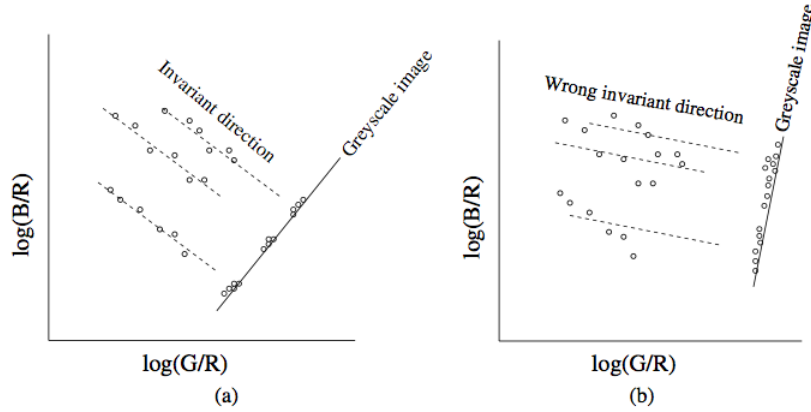


Figura 4.9: (a) Calibrazione con angolo corretto (b) Calibrazione con angolo errato.

Il procedimento consiste nell'analizzare gli istogrammi derivati da più immagini I di calibrazione. Partendo da una immagine a colori, ne si calcola la proiezione sull'asse l_α , utilizzando il procedimento appena visto, per ogni angolo $\alpha \in [-90, 89]$, ottenendo una immagine I_α in scala di grigi. Poniamo ε_α l'entropia dell'immagine I_α , che è definita utilizzando l'istogramma di tale immagine H_α come in equazione 4.6

$$\varepsilon_\alpha = -\sum_i^L H_\alpha(i) \log(H_\alpha(i)) \quad (4.6)$$

dove L è il numero di bin dell'istogramma H_α . A questo punto possiamo ottenere una distribuzione dei valori di entropia al variare dell'angolo α . L'idea di base (esemplificata in figura 4.9) è il fatto che utilizzando un valore di α errato, cioè $\alpha \neq \theta$, colori con componente cromatica principale simile vengono proiettati in modo sparso sull'asse l_α contribuendo a bin differenti di H_α che alla fine va ad aumentare il valore di ϵ_α . Con il valore dell'angolo corretto, cioè $\alpha = \theta$, colori con componente cromatica principale simile corrispondono allo stesso bin di h_α per cui ci si aspetta un valore minore di ϵ_α . Tra i 180 istogrammi ottenuti quindi si ricerca quello che mostri il valore di entropia più basso e l'angolo che lo ha generato viene chiamato θ .

Questo tipo di calibrazione si basa solo sul contenuto di immagini in arrivo dalla telecamera e perciò può essere adattato per l'autocalibrazione a bordo veicolo. Tuttavia l'analisi di questi istogrammi rivela una certa instabilità in base al contenuto dell'immagine che si ripercuote inficiando una misura coerente dell'angolo di calibrazione. Per ovviare a questo inconveniente è possibile eseguire la calibrazione su molte immagini (circa 15) e poi fare una media degli angoli ottenuti. Si capisce quindi che uno dei passi chiave per la riuscita dell'algoritmo è l'individuazione di una corretta bin-width usata per ottenere H_α . Per calcolare la larghezza dei bin si fa quindi affidamento sulla regola di Scott, che perciò viene posta al valore $3.5N^{-(1/3)}std(I_\alpha)$ dove $std()$ sta per deviazione standard e N è il numero di pixel di I_α . Inoltre viene ridotta l'influenza degli outliers considerando solo il 90% di mezzo del range di valori di I_α durante il calcolo di H_α . Quindi in questo caso, N è il numero di pixel dentro questa percentuale.

Una volta ottenuto l'angolo di calibrazione corretto, è possibile creare un'immagine in scala di grigi proiettando le coordinate cromatiche logaritmiche di ogni pixel su un asse con la pendenza ottenuta, secondo la seguente legge di proiezione riportata in 4.7

$$\begin{aligned} I &= r_{LCD}\cos(\theta) + b_{LCD}\sin(\theta) \\ r_{LCD} &= \log(R/G) \\ b_{LCD} &= \log(B/G) \end{aligned} \tag{4.7}$$

In seguito i valori ottenuti di I devono essere riscaldati in modo da variare in $[0,255]$ e poter essere considerati, ai fini della visualizzazione, dei valori di grigio per un'im-

immagine a una singola banda cromatica.

Ogni valore di grigio così ottenuto rappresenta quindi una "mappatura" dei colori presenti nell'immagine RGB iniziale, in cui ogni colore "intrinseco" dell'immagine originale (cioè non il colore percepito dal sensore che è soggetto a influenze dell'illuminazione, ma il vero colore della superficie inquadrata) viene mappato appunto su un particolare valore di grigio. L'immagine ottenuta minimizza le influenze dell'illuminazione sulle componenti cromatiche principali delle superfici inquadrate eliminando in parte effetti indesiderati dovuti all'illuminazione, come per esempio le ombre.

4.2.2 Algoritmo

L'algoritmo di rilevamento ed inseguimento di un veicolo è diviso in due step fondamentali:

- Addestramento: in questa fase si attende un input dall'utente e si addestra il tracker all'inseguimento del veicolo;
- Inseguimento: in questa fase, per ogni frame, si insegue il veicolo Leader nel sistema di riferimento dell'immagine, per poi ricavarne la posizione fondendo i dati visivi con i dati del sensore Lidar.

La fase di addestramento inizia quando l'utente, tramite l'HMI del veicolo intelligente, indica quale veicolo si deve seguire. Dato l'input, viene ritagliato all'interno dell'immagine un *bounding box*, cioè un rettangolo che comprende, nel miglior caso solo e soltanto, il veicolo all'interno dell'immagine in analisi. Sui dati dell'immagine interni al bounding box vengono eseguite due operazioni fondamentali:

- Inizializzazione delle *features*;
- Generazione del modello di colore.

Inizializzare le *features*, cioè i punti caratteristici di una immagine, significa creare un insieme di punti dell'immagine che siano distinguibili e unici, in modo da poterli

ritrovare nei frame successivi. Per compiere questo compito sono stati proposti moltissimi metodi [62] [63] [64], l'approccio scelto in questa applicazione è basato su *Speed-Up Robust Features* (SURF) [65]. SURF è un veloce e robusto algoritmo per ricerca e la comparazione dei punti caratteristici delle immagini. Così come accade in SIFT [63], SURF seleziona i punti d'interesse partendo dalle caratteristiche salienti dello *scale-space* e successivamente costruisce le caratteristiche locali basandosi sulla distribuzione del gradiente dell'immagine. La caratteristica più interessante di

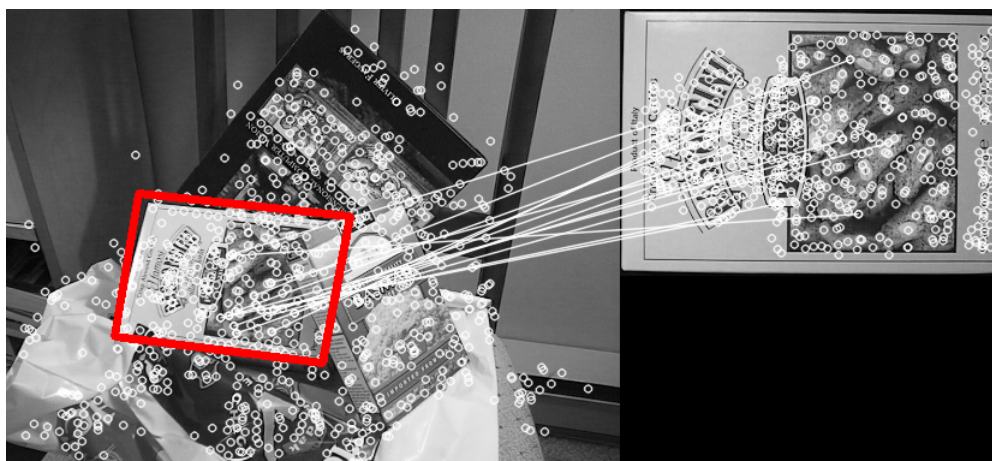


Figura 4.10: Esempio di funzionamento di SURF: invarianza alle trasformazioni geometriche e fotometriche.

questo tipo di approccio sta nella velocità del calcolo degli operatori differenziali nello *scale-space*, basati sulle immagini integrali e sui *Box Filters* [66], che permettono il calcolo ed il *matching* delle features in real-time. Proprio grazie alla possibilità di esecuzione in real-time è possibile utilizzare SURF per il rilevamento delle features in applicazione come l'inseguimento ed il riconoscimento di oggetti in movimento nelle immagini. Le performance di SURF sono garantite, com'è possibile vedere in figura 4.10, dall'invarianza alle trasformazioni geometriche (traslazione, rotazione e scala) e fotometriche.

Per quanto riguarda la generazione del modello di colore si parte dalla conversione dell'immagine contenuta nel bounding box nello feature space di cui si è parlato nel precedente paragrafo, come esemplificato in figura 4.11. Ponendo come vincolo

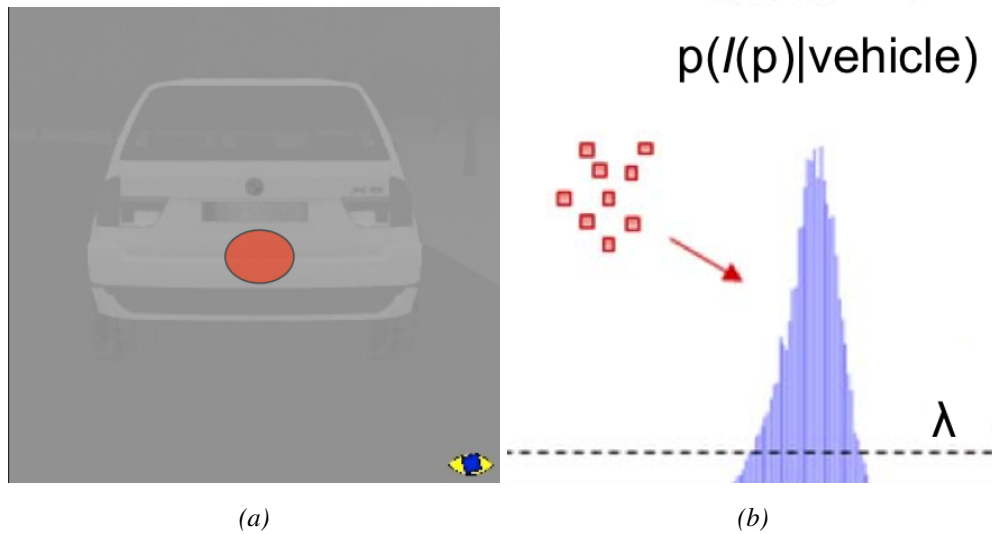


Figura 4.11: (a) Immagine intrinseca di partenza (b) Generazione del modello di colore.

che il punto centrale nel secondo terzo in basso dell'immagine convertita contenga il veicolo in oggetto, è possibile creare un modello di colore partendo da un intorno circolare di questo punto. In particolare, si utilizza, come visibile in figura 4.11, un istogramma normalizzato dei punti appartenenti all'intorno per generare il modello statistico non parametri $p(I(p)|vehicle)$, il quale, per ogni livello di grigio dell'immagine intrinseca definisce la probabilità di appartenere al veicolo.

Terminata la fase di apprendimento, che può durare anche più di un singolo frame, si passa alla fasi di inseguimento vero e proprio del veicolo. L'algoritmo di inseguimento del veicolo è composto dai seguenti passi:

1. Matching delle features: il primo passo consiste nell'effettuare il matching delle features appartenenti al veicolo nel nuovo frame, cioè ricercare le vecchie features all'interno della nuova immagine, per poterne valutare lo spostamento.

2. Filtraggio delle features: l'elenco delle features viene filtrato dalle features con spostamento rispetto al frame precedente non conforme all'andamento generale utilizzando RANSAC. Gli *outliers* eliminati appartengono o ad un errato match oppure ad una feature non appartenente al veicolo;
3. Calcolo della matrice di spostamento: dato l'insieme delle features filtrate, viene calcolata la matrice affine che descrive lo spostamento effettuato dalle features per passare dal vecchio frame al nuovo frame;
4. Spostamento del bounding box: utilizzando la matrice di spostamento si applica la trasformazione $[x'_i, y'_i]^T = [R_\theta | t][x_i, y_i]^T$ ad ogni vertice del bounding box in modo da ottenere un nuovo bounding box contenente il veicolo;
5. Rifinitura del bounding box: utilizzando il modello di colore definito in precedenza, si classificano tutti i pixel appartenenti al bounding box come appartenenti o meno al veicolo, considerando anche i pixel limitrofi al bounding box stesso. La media e varianza dei pixel classificati come positivi definiscono la nuova posizione e dimensione del veicolo in coordinate immagine.
6. Fusione sensoriale: dati i punti laser segmentati in oggetti, si cerca quindi quale oggetto nei dati laser rappresenta il veicolo inseguito nell'immagine; la relazione da soddisfare è $L_i \in C_V \leftrightarrow f(x_{Li}) \in I_{bbox}$. In sostanza, è definito come veicolo da inseguire l'oggetto laser i cui punti proiettati in coordinate immagine sono interamente inclusi nel bounding box del veicolo. Una volta trovato L_i , si utilizzano i dati sulla distanza e posizione dell'oggetto laser come distanza e posizione del veicolo da inseguire.

4.2.3 Risultati

L'algoritmo di inseguimento di un veicolo è stato provato sia con video presi attraverso i veicoli del progetto VIAC, come raffigurato in figura 4.12, sia con sequenze simulate, attraverso il simulatore Prescan descritto in appendice, come visibile in figura 4.13.



Figura 4.12: Inseguimento veicolo: applicazione a video VIAC.

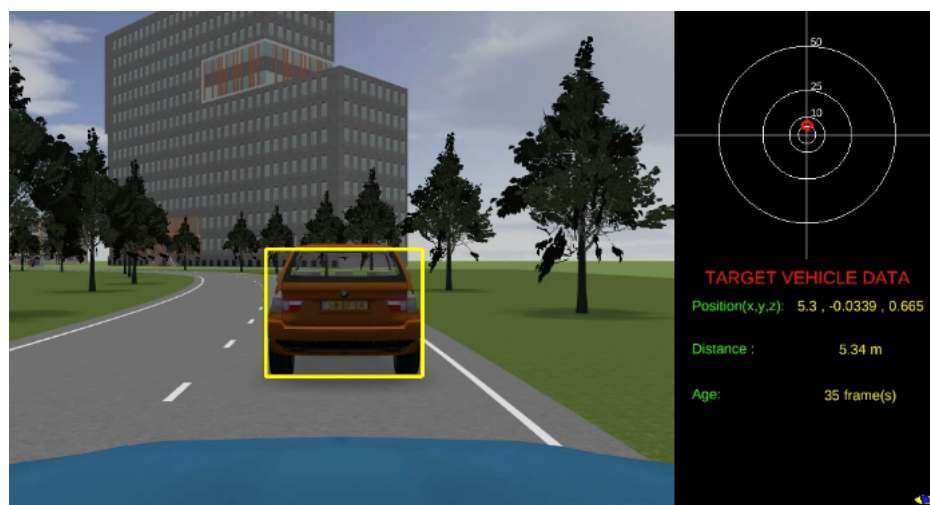


Figura 4.13: Inseguimento veicolo: applicazione a situazioni simulate.

Come è possibile notare da entrambe le figure, l'algoritmo ha un funzionamento ottimo in ogni situazione di luminosità e per forti spostamenti da frame a frame del

veicolo. In particolare, l'utilizzo delle immagini intrinseche da una grande robustezza ai cambiamenti di illuminazione, mentre il particolare inseguimento implementato attraverso la definizione dello spostamento affine delle features da al sistema una grande reattività anche per forti spostamenti reciproci dei veicoli leader e follower. Infine, l'utilizzo della fusione con dati laser fa sì che il sistema sia molto preciso nella rilevazione della distanza e che, grazie alla particolare tecnologia utilizzata, questa precisione non degradi all'aumentare della misura.

L'utilizzo del simulatore Prescan, e la conseguente presenza di un *ground truth* su cui calcolare le performance dell'algoritmo, ha permesso di estrarre dati molto precisi per quanto riguarda le performance dell'algoritmo nelle diverse situazioni meteorologiche e di luminosità; i dati estratti sono presentati in tabella 4.1.

	Frame	Correct Detection	False positive	False Negative
Sole	5000	94	1	5
Variabile	5000	89	4	7
Pioggia	5000	96	1	3
Nebbia	5000	85	9	6

Tabella 4.1: Risultati con diverse situazioni meteorologiche dell'algoritmo di inseguimento veicolo.

I dati presentati evidenziano un comportamento del sistema generalmente ottimo, con percentuali di correct detection sempre superiori al 90%. I fattori che causano una degradazione delle prestazioni del sistema sono principalmente le situazioni di tempo variabile, in cui probabilmente i forti e improvvisi cambiamenti di luminosità causano una perdita di precisione del sistema, e le situazioni di nebbia in cui lo *sharpening* dato dalla particolare condizione atmosferica causa l'impossibilità di mantenere una precisione ottimale.

Capitolo 5

Conclusioni

In questa tesi di dottorato si è preso in analisi il problema di una architettura per veicoli intelligente che potesse supportare diversi livelli di automazione ed, inoltre, si è implementato un paradigma di guida altamente automatizzato in grado di far cooperare veicolo e conducente nell'attività di guida. Per ogni argomento affrontato si è cercato di implementare nuovi algoritmi e strutture software che permettessero di migliorare le prestazioni dei sistemi esistenti.

Per quanto riguarda lo sviluppo di un nuovo paradigma di guida altamente automatizzata si è inizialmente presentato lo stato dell'arte in questo campo, soprattutto relativamente alla filosofia base di guida di un veicolo. Successivamente si è presentata la proposta di guida altamente automatizzata (o a livello di manovra), detta Digital Driving, che promuovendo l'interazione fra conducente e veicolo dal livello di controllo al livello di manovra permette la guida del veicolo attraverso elementi discreti detti *behaviours*. L'utilizzo del Digital Driving permette il miglioramento delle condizioni di sicurezza di un veicolo poiché permette di limitare l'intervento umano all'interno della guida, riducendo in questo modo i rischi introdotti dal fattore umano; allo stesso tempo, mantenendo il guidatore nel ciclo percezione-azione, facilita il compito del veicolo intelligente promuovendo e regolando l'interazione fra il veicolo e il conducente, che in questo modo possono cooperare per la riuscita dell'attività di guida. Come ultimo vantaggio di questo approccio si ha il raggruppamento di tutti gli

ADAS all'interno di una unica HMI, facilitando e semplificando il compito del guidatore grazie ad una forte riduzione del carico di lavoro mentale richiesto dall'attività di guida.

Rispetto allo stato dell'arte il Digital Driving si può avvantaggiare dei seguenti fattori:

- Completezza: il catalogo delle manovre proposto risulta essere completo, in quanto può coprire tutte le possibili soluzioni di guida. Questo è stato reso possibile dall'analisi dei dati ottenuti dal progetto VIAC.
- Compattezza: il catalogo delle manovre, grazie all'astrazione introdotta dalla presenza dei cosiddetti behaviours, risulta essere particolarmente compatto (solo 7 manovre) a differenza dello stato dell'arte analizzato. Un catalogo delle manovre compatto è di vitale importanza per aumentare la *user acceptance* del sistema implementato.
- Policy di gestione dei conflitti: il Digital Driving, a differenza dei principali elementi dello stato dell'arte, cerca di regolamentare gli eventuali conflitti fra conducente e veicolo intelligente attraverso una policy di gestione dei conflitti multi-livello che ha lo scopo di risolvere gli eventuali conflitti allo stesso livello di astrazione in cui i conflitti nascono.

Una delle possibili incognite del Digital Driving riguarda la possibile *user acceptance* di questi tipi di sistemi di guida: la guida a livello di manovra risulta infatti essere un'attività molto diversa dalla normale guida di un veicolo, anche se il largo utilizzo, da parte degli utenti, degli ADAS di cui sono dotati i veicoli odierni fa ben sperare in questo senso. Nel prossimo futuro sarà in ogni caso necessario effettuare uno studio dell'accettazione degli utenti per meglio definire le HMI del sistema in modo da rendere il sistema il più possibile accettato dagli utenti.

Per quanto riguarda l'architettura si è progettata un'architettura software per veicoli intelligenti in grado di soddisfare i requisiti del progetto OFAV ed, allo stesso tempo, di migliorare le preesistenti architetture dello stato dell'arte. I vantaggi rispetto quest'ultime sono i seguenti:

- Generalità: l'architettura è stata progettata per coprire tutte le possibili attività di un generico veicolo intelligente, a differenza dei principali casi dello stato dell'arte in cui l'architettura era focalizzata nella risoluzione di determinati casi (come i percorsi urbani nel caso delle architetture derivanti dalla Urban Challenge). In aggiunta l'architettura oggetto di questa tesi può essere utilizzata anche in ambiti diversi da quello strettamente automotive, come per esempio in ambito agricolo o minerario.
- Modularità e scalabilità: l'intera architettura è stata progettata in modo da garantire la massima modularità possibile; inoltre l'utilizzo del VLBUS all'interno dell'architettura permette di svincolare completamente ogni modulo dal tipo e dalla posizione del device di esecuzione.
- Integrazione livelli di automazione: nell'architettura sono stati integrati tutti i livelli di automazione visti nel capitolo 1, a differenza dello stato dell'arte in cui di architettura in architettura era stato implementato un solo livello di automazione. In particolare, l'integrazione è stata curata in modo da garantire il massimo riutilizzo di componenti attraverso diversi livelli di automazione: questo è stato realizzato mantenendo la stessa struttura software in tutti i livelli di automazione e sostituendo di livello in livello le attività eseguite dal conducente umano con componenti software dedicate.
- Apertura: l'architettura progettata utilizza solamente componenti aperti e non fa utilizzo di dati proprietari, a partire dalle mappe che appartengono al progetto "OpenStreet Map"; il particolare tipo di layer di pianificazione implementato permette di poter utilizzare questo tipo di mappe gratuite ma molto meno precise di quelle a disposizione di altri progetti.
- Fusione Sensoriale multilivello: nell'architettura è possibile effettuare la fusione sensoriale su diversi livelli, dal basso livello della fusione interna ad ogni modulo percettivo alla fusione *track-to-track* effettuata internamente al WPS.

Gli studi che sono stati portati a termine e gli algoritmi sviluppati hanno permesso un miglioramento delle prestazioni dei sistemi preesistenti, oppure l'aggiunta di nuove

Appendice A

Considerazioni legali sui sistemi di guida automatica

A.1 Leggi regolatrici per i veicoli odierni

La prima distinzione da effettuare riguarda le aree di traffico ad accesso limitato e le aree di traffico ad accesso pubblico: i codici di regolamentazione del traffico sono limitati ai terreni aperti al traffico pubblico; di conseguenza quanto diremo sotto vale solo per le aree aperte al pubblico mentre per le aree private le regole riportate non sono applicabili. Detto questo, la regolamentazione riguardante il traffico cambia di nazione in nazione anche in maniera sostanziale, basti considerare le differenze fra nazioni con circolazione a destra e le nazioni con circolazione a sinistra. Nonostante ciò, per diminuire il più possibile le differenze fra nazione e nazione, soprattutto a riguardo di nazioni confinanti, le leggi sulla guida di un veicolo sono soggette a trattati internazionali che richiedono ai vari governi locali di modificare i loro codici della strada coerentemente con gli accordi internazionali.

La convenzione di Vienna sul traffico stradale è un trattato internazionale riguardante la regolamentazione della guida dei veicoli ed il traffico in generale. Nonostante sia stato negoziato nel 1968 sotto la UNECE (United Nations Economic Commission of Europe), la Convenzione di Vienna è tuttora valida e ratificata recentemente (13

Febbraio 2010) da più di 70 nazioni di tutto il mondo. Essendo un trattato internazionale, non causa cambiamenti immediati nelle legislazioni locali, ma gli stati ratificatori dovranno adeguarsi alla Convenzione nel passare degli anni.

Nell'analisi seguente sugli effetti delle leggi regolatrici sulla conduzione di un veicolo, l'attuale situazione legale verrà confrontata con i possibili effetti dei sistemi di guida intelligente nella guida di un autoveicolo; in particolare oggetto di analisi sarà il potenziale di supporto e aiuto che un veicolo intelligente può fornire ad un guidatore. Naturalmente, le norme giuridiche del passato non potevano prendere in considerazione lo sviluppo sulla strada dei veicoli intelligenti, il cui sviluppo era considerato imprevedibile al momento della negoziazione / stipulazione delle norme di circolazione. Tuttavia la normativa deve rimanere applicabile in caso di veicoli intelligenti in modo da non incorrere in incoerenze e contraddizioni che ostacolano l'attuazione e quindi diminuisca i vantaggi per la sicurezza del traffico e aumentando così i rischi combinati con la commercializzazione di tali veicoli.

L'esistenza ed il ruolo del guidatore sono chiamaramente affermati dagli estratti dei seguenti articolo presi dalla convenzione di Vienna:

- L'art. 1 definisce come "conducente": *"...ogni persona che assume la guida di un veicolo, autoveicolo od altro (compresi i velocipedi) o che, su di una strada, guida del bestiame, isolato o in greggi, o degli animali da tiro, da soma o da sella;"*;
- Art. 7 par. 1 sostiene: *"Gli utenti della strada debbono evitare ogni comportamento che possa costituire un pericolo o un ostacolo per la circolazione, mettere in pericolo le persone o provocare un danno alle proprietà pubbliche o private."*;
- Art.8 par.1: *"Ogni veicolo in movimento o ogni complesso di veicoli in movimento deve avere un conducente."*;
- Art.8 par.5: *"Ogni conducente deve avere costantemente il controllo del proprio veicolo o deve poter guidare i propri animali"*;

- Art.13 par.1: *"Ogni conducente di veicolo deve, in ogni circostanza, restare padrone del proprio veicolo, in modo da potersi conformare alle esigenze della prudenza e da essere costantemente in grado di effettuare tutte le manovre che gli competono.[...]"*;
- Art. 14 par.1: *"Ogni conducente che vuole eseguire una manovra, come uscire da una fila di veicoli in sosta o entrarvi, spostarsi a destra o a sinistra sulla carreggiata, svoltare a sinistra o a destra per percorrere un'altra strada o per entrare in una proprietà fiancheggiante la strada, deve cominciare ad eseguire tale manovra soltanto dopo essersi assicurato che può farlo senza rischiare di costituire un pericolo per gli altri utenti della strada che lo seguono, lo precedono o stanno per incrociarlo, tenuto conto della loro posizione, della loro direzione e della loro velocità[...]"*

Può quindi essere concluso che in termini di conduzione di un veicolo è un obbligo del conducente rimanere in carica del proprio veicolo ed intervenire in caso di comportamenti non corretti del veicolo (Art. 8 e 13 di cui sopra). Questo quindi implica la necessaria presenza di un sistema di sovrascrittura dei comandi del veicolo in modo da permettere l'intervento del conducente. D'altra parte questi articoli assumono anche che il guidatore deve essere permanentemente concentrato nell'attività di guida in modo da poter reagire se necessario; questa stessa regola sull'attenzione del conducente può essere derivata dagli articoli 7 e 14 della Convenzione di Vienna. La condizione, come già analizzato nel passato, risulta veritiera anche per i DAS, poichè sistemi in cui il guidatore non ha la possibilità di sovrascrivere i comandi di un veicolo intelligente contrasterebbero con la regolamentazione in materia; in particolare quindi uno dei requisiti di un veicolo intelligente, anche se completamente autonomo, è la possibilità, da parte del conducente, di sovrascrivere in ogni momento il comportamento del veicolo.

Il secondo importato principio emerso dall'analisi della Convenzione di Vienna riguarda il comportamento del conducente, che deve essere sempre concentrato nell'attività di guida. Ovviamente, al tempo in cui la convenzione è stata concepita, la guida autonoma non era nemmeno stata concepita, di conseguenza risulta evidente

come il guidatore non potesse distrarsi nemmeno per periodi molto brevi da tale attività. Inoltre, anche con il sopraggiungere di sistemi di aiuto alla guida, i due principi soprastanti non sono stati messi in discussione, poichè come risulta evidente anche dal termine stesso, essi sono stati concepiti per assistere il guidatore nell'attività di guida, senza però sostituirsi ad esso. Risulta quindi evidente come, anche nel caso dei DAS, il conducente rimane in una posizione di superiorità rispetto all'automazione e di conseguenza non ha la possibilità di spostare la propria attenzione dall'attività di guida.

Per quanto riguarda gli ADAS si può affermare che, similmente a quanto detto per i DAS, essi possono funzionare solo a stretto regime di cooperazione con il conducente. Tipicamente infatti gli ADAS forniscono solo una parte molto limitata di automazione, ad esempio fornendo il controllo longitudinale (ACC) mentre il conducente effettua il controllo laterale (o viceversa). Di conseguenza, questo tipo di sistemi richiedono una costante sorveglianza da parte del guidatore e le correzioni necessarie sono effettuate direttamente sovrascrivendo i comandi del sistema intelligente.

Un comportamento diverso può essere ravvisato in alcuni tipi di DAS, come ad esempio ABS (Antilock bracking system) che effettivamente interviene riducendo la pressione sul freno applicata dal conducente, senza dargli la possibilità di sovrascrivere il comando. Questo, dal punto di vista della regolamentazione, è possibile poichè il sistema è visto come un miglioramento dell'attività di frenata, e non come una automazione vera e propria. Similmente, si può analizzare il caso di ESC (Electronic Stability Control): durante la guida questo sistema monitora continuamente lo sterzo e la direzione del veicolo. La direzione del veicolo e la direzione voluta dal guidatore (ottenuta come conversione dell'angolo di sterzo), sono le basi per l'intervento dell'ESC: esso infatti interviene quando le due direzioni sono diverse, cioè quando sta avvenendo probabilmente una perdita di controllo del veicolo da parte del conducente, correggendo la sterzata in modo da stabilizzare il veicolo. Anche in questo caso il comando non è sovrascrivibile (se non disattivando il sistema), ma anche in questo caso il DAS è visto come un semplice miglioramento dell'attività di guida poichè rimane una semplice attuazione della volontà del conducente (l'interven-

to infatti è limitato all'angolo di sterzo comandato dal conducente). L'intervento di questo tipo di automazione è considerato legale anche a causa del fatto che avviene solo in situazioni di emergenza in cui la percezione è certa del fatto che il problema non sarebbe più evitabile in altro modo.

A.2 Sistemi ad elevata automazione

I sistemi ad elevata automazione, come il Digital Driving descritto in questa tesi, sono dotati di un maggiore grado di automazioni e di minori limiti di sistema in confronto agli ADAS attualmente sul mercato. Di conseguenza, il controllo diretto continuativo da parte del conducente non è più necessario ma i limiti del sistema sono tali per cui potrebbe essere necessaria una supervisione del guidatore all'intera attività di guida, soprattutto in caso di situazioni di traffico molto difficoltose; di conseguenza il sistema deve rimanere sovrascrivibile da parte del conducente in ogni momento. Dal punto di vista del codice della strada la guida ad elevata automazione è simile alla guida con ADAS: il sistema di guida intelligente infatti anche in questo caso richiede la presenza di un conducente, almeno per esigenze di supervisione sull'attività del veicolo intelligente. Le due questioni principali, dal punto di vista legislativo, riguardano ancora una volta la possibilità di sovrascrivere i comandi del veicolo e l'attenzione del conducente nell'attività di guida. Per quanto riguarda la possibilità di sovrascrittura dei comandi essa dovrebbe essere garantita anche a questo livello di automazione, poichè se in alcuni casi la guida del veicolo potrà essere affidata totalmente all'automazione (guida detta "hands-off", cioè con le mani non poste sul volante) in altri casi particolari situazioni di pericolo o di traffico richiederanno al conducente di intervenire direttamente nel controllo del veicolo. Di conseguenza, la possibilità di sovrascrivere i comandi è garantita anche a questo livello.

Dal punto di vista legislativo potrebbe introdurre più problemi la necessità di prestare costante attenzione nell'attività di guida: la guida "hands-off" per natura porterebbe alla distrazione del conducente, che proprio grazie a questo tipo di automazione potrebbe impiegare il proprio tempo in una qualche diversa attività. Allo stesso tempo, questo tipo di guida potrebbe ritardare la reazione del conducente, proprio a

causa della mancanza di concentrazione nella guida; di conseguenza per assicurarne la fattibilità legale del veicolo intelligente sarà necessario implementare un qualche tipo di sistema che consenta di monitorare e se necessario stimolare l'attenzione del conducente nell'attività di guida.

A.3 Sistemi completamente autonomi

Un veicolo intelligente completamente automatico rappresenta il passaggio successivo al veicolo ad elevata automazione. In questo caso il veicolo dovrebbe avere capacità tali da permettergli di essere totalmente indipendente dal conducente, che in questo caso diviene un semplice passeggero. Come sopra, le due principali problematiche riguardano la possibilità di sovrascrivere le azioni decise dal veicolo e le possibili distrazioni del conducente. A differenza del caso di sistemi ad elevata automazione, in caso di un sistema completamente autonomo il conducente viene escluso dal ciclo percezione azione, poichè il sistema dovrebbe avere capacità tali garantire la sicurezza di se stesso e dei passeggeri che trasporta. Per questo motivo non si può più parlare di conducente ma bensì di passeggero, e per questo motivo non ha più senso parlare nè di possibilità di sovrascrittura nè di attenzione del conducente, poichè semplicemente un conducente a questo livello non esiste. Di conseguenza l'attuale normativa stradale non consentirebbe la circolazione di un veicolo completamente autonomo e per questo motivo è necessario un'aggiornamento di tale normativa che consideri gli ultimi sviluppi della tecnologia in ambito automobilistico.

Appendice B

Testing di un veicolo intelligente

L'attività di *testing* e *debugging* di un veicolo intelligente, cioè l'analisi delle performance dei vari sistemi, unita all'attività di ricerca di eventuali problemi ed errori nella parte hardware o software, è un'attività molto complessa. I motivi di tale complessità sono già stati evidenziati nel capitolo 3, ed in sintesi sono:

- Unicità: il veicolo intelligente normalmente è una singola risorsa alla quale tutti i componenti del gruppo necessitano di accedere. Raramente si ha a disposizione un secondo veicolo, clone del primo, su cui effettuare test in parallelo.
- Complessità dei Test: uno dei problemi principali risulta essere la complessità generale del problema. Questo si riflette anche nella fase di test del sistema in quanto in caso di problemi, data la complessità della struttura, risulta difficile disaccoppiare le fonti di ogni problema.

Per questi motivi si è deciso di utilizzare un simulatore per analizzare ed effettuare il *debug* dell'architettura software e del paradigma di Digital Driving. Il simulatore scelto è il software Prescan, della Tass-Safe. Prescan è uno strumento software che è utilizzabile per lo sviluppo di ADAS basati su tecnologie sensoriali come radar, laser, telecamere o GPS; inoltre Prescan è utilizzabile per sperimentazione di tecnologie di comunicazione Vehicle-2-Vehicle(V2V) e Vehicle-2-Infrastructure. Prescan

è composto da diversi moduli, tra i quali una interfaccia grafica attraverso la quale è possibile costruire l'ambiente di simulazione ed impostare tutti i parametri necessari alla simulazione stessa, tra cui i parametri dei sensori e dei veicoli sulla scena; la simulazione vera e propria avviene all'interno di Matlab/Simulink, il che rende possibile aggiungere facilmente un sistema di controllo oltre a rendere possibile l'utilizzo di modelli dinamici di veicoli creati con altri software (come CarSim). Un ulteriore modulo è costituito dal visualizzatore 3D, il quale ha il compito di visualizzare e registrare ciò che avviene nella simulazione da diversi punti di vista. Opzionalmente è possibile connettere il software con altri tools come dSPACE ControlDesk, per poter importare dati da fonti non simulate. L'implementazione di una simulazione

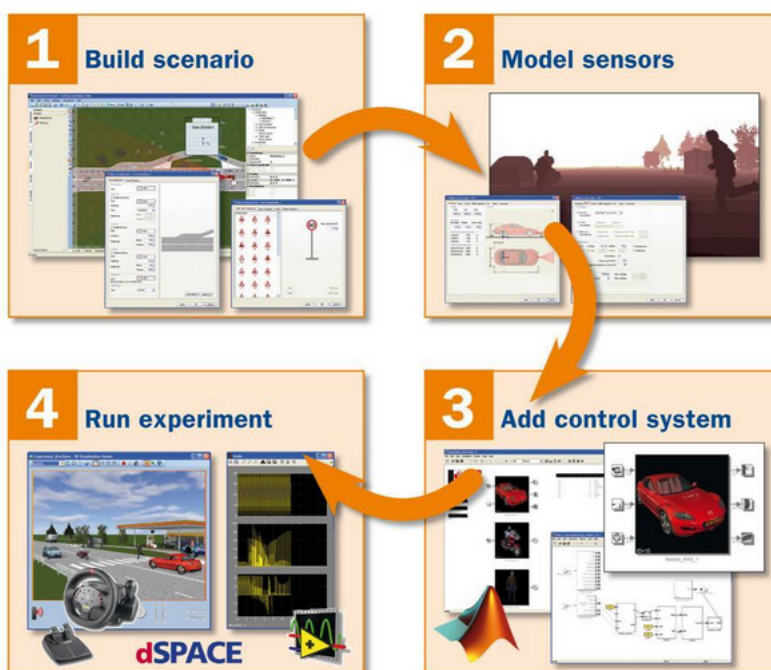


Figura B.1: Passi fondamentali nell'implementazione di una simulazione in Prescan (image credit www.tass-safe.com).

all'interno di Prescan è costituita dai seguenti passi, visualizzati in figura B.1:

- **Costruzione dello scenario:** la GUI permette la costruzione e modifica del mondo all'interno del quale avviene la simulazione. Attraverso la GUI è possibile accedere ad una libreria con diversi segmenti di strada, componenti infrastrutturali (alberi, edifici, cartelli stradali etc.), attori (veicoli, camion, biciclette e pedoni), condizioni atmosferiche (sole, pioggia etc.) e fonti luminose (compresa la posizione ed intensità del sole). Allo stesso modo è possibile utilizzare segmenti stradali reali prendendo i dati da OpenStreetMap, Google Earth e similari.
- **Modellazione dei sensori:** in questa fase è possibile, sempre attraverso la GUI, dotare gli attori presenti sulla scena di vari sensori, tra cui telecamere, laser e GPS.
- **Aggiungere un sistema di controllo:** una volta creato completamente lo scenario è possibile accedere Matlab/Simulink e aggiungere un controllo ai veicoli simulati, oppure cambiare la loro dinamica importando dati da altri software dedicati, come per esempio CarSim o veDyna.
- **Esecuzione dell'esperimento:** il visualizzatore 3D permette di visualizzare ed analizzare i risultati dell'esperimento mentre esso viene eseguito. Oltre alle sopracitate capacità, tra cui la possibilità di vedere e registrare da diversi punti di vista contemporaneamente, Prescan permette la connessione a software come ControlDesk e LabView per l'esecuzione automatica di esperimenti e per l'esecuzione di simulazione HIL (Hardware In the Loop).

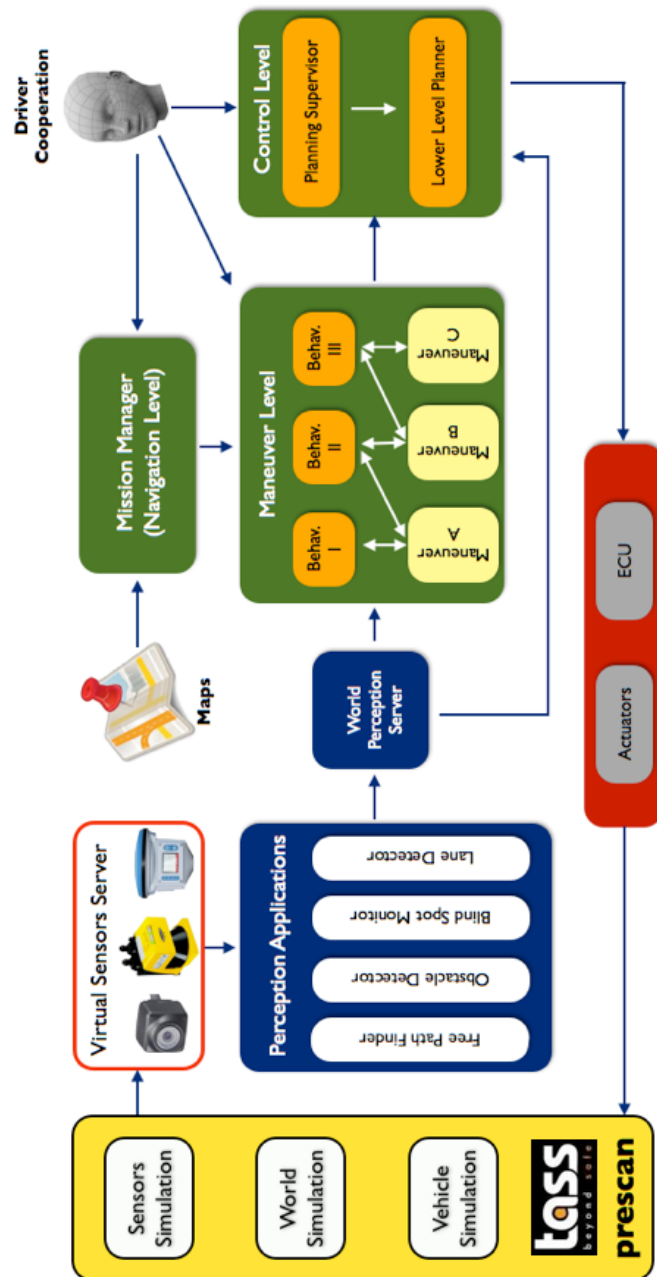


Figura B.2: Testing dell'architettura: integrazione del simulatore.

L'approccio utilizzato nell'integrazione del simulatore all'interno dell'architettura è visualizzato in figura B.2; in sostanza si è scelto di scrivere dei moduli software che consentissero di mantenere invariata l'intera architettura, sostituendo al mondo reale il mondo simulato all'interno di Prescan. Questo approccio porta il vantaggio principale di annullare le differenze, da un punto di vista software, fra la simulazione ed il caso reale con il conseguente vantaggio di non dover modificare sezioni di software nel passaggio da una situazione all'altra. Per quanto riguarda la percezione, a connessione fra il simulatore e l'architettura è stata fatta scrivendo due moduli software per ogni tipo di sensore: un primo modulo, interno a Simulink, si occupa di leggere i dati dal simulatore, impacchettarli in modo che siano comprensibili dal software interno all'architettura e successivamente spedirli utilizzando VLBUS come mezzo di comunicazione. Dall'altro lato, internamente all'architettura, si è scritto un driver generico in grado di leggere dati da VLBUS, driver che poi è stato specializzato in modo diverso per ogni sensore da cui si volevano ricevere dati. Per quanto riguarda la parte di attuazione si è fatto l'opposto: un driver interno all'architettura si occupa di scrivere dati su determinate linee VLBUS mentre un modulo interno a Simulink si occupa di leggere i dati da VLBUS e trasmettere i comandi contenuti al simulatore Prescan.

Bibliografia

- [1] European Road Safety Observatory. Annual statistical report, 2008.
- [2] Rune Elvik. Speed and road safety: Synthesis of evidence from evaluation studies. *Transportation Research Record: Journal of the Transportation Research Board*, 1908:59–69, 2005.
- [3] Vicki L. Neale, Thomas A. Dingus, Sheila G. Klauer, Jeremy Sudweeks, and Michael Goodman. An overview of the 100-car naturalistic study and findings. national highway traffic safety administration and virginia tech transportation institute. www.nhtsa.dot.gov/staticfiles/dot/nhtsa/nrd. page 787, 2005.
- [4] European Commission Directorate-General for Energy and Transport. Energy and transport in figures - statistical pocketbook 2010, 2010.
- [5] Deborah Braid, Alberto Broggi, and Gary Schmiedel. The TerraMax Autonomous Vehicle concludes the 2005 DARPA Grand Challenge. In *Procs. IEEE Intelligent Vehicles Symposium 2006*, pages 534–539, Tokyo, Japan, June 2006.
- [6] Alberto Broggi, Andrea Cappalunga, Claudio Caraffi, Stefano Cattani, Stefano Ghidoni, Paolo Grisleri, Pier Paolo Porta, Matteo Posterli, and Paolo Zani. TerraMax Vision at the Urban Challenge 2007. *IEEE Trans. on Intelligent Transportation Systems*, 11(1):194–205, March 2010.
- [7] Joachim Irion Maxime Flament Matthias Schulze, Tapani Makinen and Tanja Kessel. IP_D15: Final report. Technical report, Preventive and Active Safety Applications. Technical report, Integrated Project, 2006.

- [8] A. Amditis, A. Bolvinou, U. Iurgel, A. Etemad, E. Johansson, A. Saroldi, and L. Bjelkeflo. Enhanced perception shared by automotive safety applications supporting active intervention-the interactive system overview. In *18th ITS World Congress*, 2011.
- [9] Alberto Broggi, Pietro Cerri, Stefano Ghidoni, Paolo Grisleri, and Ho Gi Jung. A New Approach to Urban Pedestrian Detection for Automatic Braking. *IEEE Trans. on Intelligent Transportation Systems*, 10(4):594–605, December 2009. ISSN: 1524-9050.
- [10] Mirko Felisa and Paolo Zani. Robust monocular lane detection in urban environments. In *Procs. IEEE Intelligent Vehicles Symposium 2010*, pages 591–596, San Diego, CA, USA, June 2010.
- [11] Yi-Liang Chen, Venkataraman Sundareswaran, Craig Anderson, Alberto Broggi, Paolo Grisleri, Pier Paolo Porta, Paolo Zani, and John Beck. TerraMax: Team Oshkosh Urban Robot. *Journal of Field Robotics*, 25(10):841–860, October 2008.
- [12] Deborah Braid, Alberto Broggi, and Gary Schmiedel. The TerraMax Autonomous Vehicle. *Journal of Field Robotics*, 23(9):693–708, September 2006.
- [13] Massimo Bertozzi, Luca Bombini, Alberto Broggi, Michele Buzzoni, Elena Cardarelli, Stefano Cattani, Pietro Cerri, Alessandro Coati, Stefano Debattisti, Andrea Falzoni, Rean Isabella Fedriga, Mirko Felisa, Luca Gatti, Alessandro Giacomazzo, Paolo Grisleri, Maria Chiara Laghi, Luca Mazzei, Paolo Medici, Matteo Panciroli, Pier Paolo Porta, Paolo Zani, and Pietro Versari. VIAC: an Out of Ordinary Experiment. In *Procs. IEEE Intelligent Vehicles Symposium 2011*, pages 175–180, Baden Baden, Germany, June 2011.
- [14] Massimo Bertozzi, Luca Bombini, Alberto Broggi, Paolo Grisleri, and Pier Paolo Porta. Camera-Based Automotive Systems. In Ahmed Nabil Belbachir,

- editor, *Smart Cameras*, pages 319–338. Springer US, October 2009. ISBN: 978-1-4419-0952-7.
- [15] Alberto Broggi, Luca Mazzei, and Pier Paolo Porta. Car-driver cooperation in future vehicles. In *Procs. Intl. Conf. on Models and Technologies for Intelligent Transportation Systems*, Rome, Italy, June 2009.
- [16] Hall J and Dudek CL. Committee on user characteristics. in: Driver performance data book update: older drivers and ivhs. *Transportation research circular*, 419, 1994.
- [17] Cristy Ho and Charles Spence. Olfactory facilitation of dual-task performance. *Neuroscience Letters*, 389(1):35 – 40, 2005.
- [18] M. Horwick and K.-H. Siedersberger. Strategy and architecture of a safety concept for fully automatic and autonomous driving assistance systems. In *Intelligent Vehicles Symposium (IV), 2010 IEEE*, pages 955 –960, june 2010.
- [19] Chris Urmson, Joshua Anhalt, Drew Bagnell, Christopher R. Baker, Robert Bittner, M. N. Clark, John M. Dolan, Dave Duggins, Tugrul Galatali, Christopher Geyer, Michele Gittleman, Sam Harbaugh, Martial Hebert, Thomas M. Howard, Sascha Kolski, Alonzo Kelly, Maxim Likhachev, Matthew McNaughton, Nick Miller, Kevin Peterson, Brian Pilnick, Raj Rajkumar, Paul E. Rybski, Bryan Salesky, Young-Woo Seo, Sanjiv Singh, Jarrod M. Snider, Anthony Stentz, William Whittaker, Ziv Wolkowicki, Jason Ziglar, Hong Bae, Thomas Brown, Daniel Demitrish, Bakhtiar Litkouhi, Jim Nickolaou, Varsha Sadekar, Wende Zhang, Joshua Struble, Michael Taylor, Michael Darms, and Dave Ferguson. Autonomous driving in urban environments: Boss and the urban challenge. *J. Field Robotics*, 25(8):425–466, 2008.
- [20] United Nations. Convention on road traffic and road signs, November 1968. Vienna, Austria.

- [21] M. Schreiber, M. Kauer, and R. Bruder. Conduct by wire - maneuver catalog for semi-autonomous vehicle guidance. In *Intelligent Vehicles Symposium, 2009 IEEE*, pages 1279 –1284, june 2009.
- [22] S. Geyer S. Hakuli, M. Kluin and H. Winner. Development and validation of manoeuvre-based driver assistance functions for conduct-by-wire with ipg carmaker. In *FISITA 2010 World Automotive Congress*, june 2010.
- [23] O. Flemisch, A. Adams, Sheila R. Conway, Ken H. Goodrich, Michael T. Palmer, and Paul C. Schutte. Nasa/tm—2003-212672 the h-metaphor as a guideline for vehicle automation and interaction, 2003.
- [24] K.H. Goodrich, P.C. Schutte, F.O. Flemisch, and R.A. Williams. Application of the h-mode, a design and interaction concept for highly automated vehicles, to aircraft. In *25th Digital Avionics Systems Conference, 2006 IEEE/AIAA*, pages 1 –13, oct. 2006.
- [25] German National Equestrian Federation. The principles of driving, 2002.
- [26] S. Geyer, S. Hakuli, H. Winner, B. Franz, and M. Kauer. Development of a cooperative system behavior for a highly automated vehicle guidance concept based on the conduct-by-wire principle. In *Intelligent Vehicles Symposium (IV), 2011 IEEE*, pages 411 –416, june 2011.
- [27] E Donges. A two-level model of driver steering behavior. *Human Factors*, 20(6):691–707, 1978.
- [28] Vislab intercontinental autonomous challenge.
<http://en.wikipedia.org/wiki/VIAC>.
- [29] Alberto Broggi, Pietro Cerri, Mirko Felisa, Maria Chiara Laghi, Luca Mazzei, and Pier Paolo Porta. The VisLab Intercontinental Autonomous Challenge: an Extensive Test for a Platoon of Intelligent Vehicles. *Intl. Journal of Vehicle Autonomous Systems, special issue for 10th Anniversary*, 2011.

- [30] Alberto Broggi, Massimo Bertozzi, Alessandra Fascioli, and Gianni Conte. *Automatic Vehicle Guidance: the Experience of the ARGO Vehicle*. World Scientific, Singapore, April 1999. ISBN 9810237200.
- [31] Luca Bombini, Stefano Cattani, Pietro Cerri, Rean Isabella Fedriga, Mirko Felisa, and Pier Paolo Porta. Test bed for Unified Perception & Decision Architecture. In *Procs. 13th Int. Forum on Advanced Microsystems for Automotive Applications*, Berlin, Germany, May 2009.
- [32] Paolo Grisleri and Isabella Fedriga. The BRAiVE platform. In *Procs. 7th IFAC Symposium on Intelligent Autonomous Vehicles*, Lecce, Italy, September 2010.
- [33] G. Struck' H.-H. Nagel, W. Enkelmann. Fhg-co-driver: From map-guided automatic driving by machine vision to a cooperative driver support. *Mathematical and Computer Modelling*, 22(47):185 – 212, 1995.
- [34] B. Pulk and R. Hynes. Unified approach to vehicle follower longitudinal control system design. In *Vehicular Technology Conference, 1978. 28th IEEE*, volume 28, pages 356 – 360, march 1978.
- [35] A. Broggi, L. Bombini, S. Cattani, P. Cerri, and R.I. Fedriga. Sensing requirements for a 13,000 km intercontinental autonomous drive. In *Intelligent Vehicles Symposium (IV), 2010 IEEE*, pages 500 –505, june 2010.
- [36] M. Bertozzi, A. Broggi, E. Cardarelli, R.I. Fedriga, L. Mazzei, and P.P. Porta. Viac expedition toward autonomous mobility [from the field]. *Robotics Automation Magazine, IEEE*, 18(3):120 –124, sept. 2011.
- [37] U. Ozguner, K. Redmill, S. Biddlestone, Ming Feng Hsieh, A. Yazici, and C. Toth. Simulation and testing environments for the darpa urban challenge. In *Vehicular Electronics and Safety, 2008. ICVES 2008. IEEE International Conference on*, pages 222 –226, sept. 2008.
- [38] U. Ozguner, K.A. Redmill, and A. Broggi. Team terramax and the darpa grand challenge: a general overview. In *Intelligent Vehicles Symposium, 2004 IEEE*, pages 232 – 237, june 2004.

- [39] Carl D. Crane. The 2005 darpa grand challenge. In *Computational Intelligence in Robotics and Automation, 2007. CIRA 2007. International Symposium on*, page nil4, june 2007.
- [40] Alberto Broggi, Paolo Grisleri, Chris Yakes, and Chris Hubert. The Oshkosh-VisLab joint Efforts on UGVs: Architecture, Integration, and Results. In *Procs. of the 146th NATO Wks. on Applied Vehicle Technology panel*, Florence, Italy, May 2007.
- [41] Michael Montemerlo, Jan Becker, Suhrid Bhat, Hendrik Dahlkamp, Dmitri Dolgov, Scott Ettinger, Dirk Haehnel, Tim Hilden, Gabe Hoffmann, Burkhard Huhnke, Doug Johnston, Stefan Klumpp, Dirk Langer, Anthony Levandowski, Jesse Levinson, Julien Marcil, David Orenstein, Johannes Paefgen, Isaac Penny, Anna Petrovskaya, Mike Pflueger, Ganymed Stanek, David Stavens, Antone Vogt, and Sebastian Thrun. Junior: The stanford entry in the urban challenge. *J. Field Robot.*, 25(9):569–597, September 2008.
- [42] Michael Montemerlo, Sebastian Thrun, Hendrik Dahlkamp, and David Stavens. Winning the darpa grand challenge with an ai robot. In *In Proceedings of the AAAI National Conference on Artificial Intelligence*, pages 17–20, 2006.
- [43] S. Matzka and R. Altendorfer. A comparison of track-to-track fusion algorithms for automotive sensor fusion. In *Multisensor Fusion and Integration for Intelligent Systems, 2008. MFI 2008. IEEE International Conference on*, pages 189–194, aug. 2008.
- [44] R.C. Luo and K.L. Su. A review of high-level multisensor fusion: approaches and applications. In *Multisensor Fusion and Integration for Intelligent Systems, 1999. MFI '99. Proceedings. 1999 IEEE/SICE/RSJ International Conference on*, pages 25–31, 1999.
- [45] J. Axelsson. Hw/sw codesign for automotive applications: challenges on the architecture level. In *Object-Oriented Real-Time Distributed Computing, 2001*.

- ISORC - 2001. Proceedings. Fourth IEEE International Symposium on*, pages 123–126, 2001.
- [46] Dmitri Dolgov and Sebastian Thrun. Autonomous driving in semi-structured environments: Mapping and planning. In *Robotics and Automation, 2009. ICRA '09. IEEE International Conference on*, pages 3407–3414, may 2009.
- [47] OpenStreetMap Foundation. Openstreetmap. Available at <http://www.openstreetmap.org>.
- [48] Dmitri Dolgov, Sebastian Thrun, Michael Montemerlo, and James Diebel. Path planning for autonomous vehicles in unknown semi-structured environments. *Int. J. Rob. Res.*, 29(5):485–501, April 2010.
- [49] D.J. Zhu and J.-C. Latombe. New heuristic algorithms for efficient hierarchical path planning. *Robotics and Automation, IEEE Transactions on*, 7(1):9–20, feb 1991.
- [50] Massimo Bertozzi and Alberto Broggi. GOLD: a Parallel Real-Time Stereo Vision System for Generic Obstacle and Lane Detection. *IEEE Trans. on Image Processing*, 7(1):62–81, January 1998.
- [51] Massimo Bertozzi, Luca Bombini, Alberto Broggi, Pietro Cerri, Paolo Grisleri, and Paolo Zani. GOLD: A Complete Framework for Developing Artificial Vision Applications for Intelligent Vehicles. *IEEE Intelligent Systems*, 23(1):69–71, January–February 2008.
- [52] Bongjoe Kim, Jongin Son, and Kwanghoon Sohn. Illumination invariant road detection based on learning method. In *Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on*, pages 1009–1014, oct. 2011.
- [53] A. Broggi, M. Bertozzi, A. Fascioli, C. Guarino Lo Bianco, and A. Piazzzi. Visual perception of obstacles and vehicles for platooning. *Intelligent Transportation Systems, IEEE Transactions on*, 1(3):164–176, sep 2000.

- [54] S. Benhimane, E. Malis, P. Rives, and J.R. Azinheira. Vision-based control for car platooning using homography decomposition. In *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, pages 2161 – 2166, april 2005.
- [55] S. Sivaraman and M.M. Trivedi. Active learning based robust monocular vehicle detection for on-road safety systems. In *Intelligent Vehicles Symposium, 2009 IEEE*, pages 399 –404, june 2009.
- [56] G. Toulminet, M. Bertozzi, S. Mousset, A. Bensrhair, and A. Broggi. Vehicle detection by means of stereo vision-based obstacles features extraction and monocular pattern analysis. *Image Processing, IEEE Transactions on*, 15(8):2364 –2375, aug. 2006.
- [57] Jonathan T. Barron and Jitendra Malik. Color constancy, intrinsic images, and shape estimation. In *ECCV (4)*, pages 57–70, 2012.
- [58] Graham D. Finlayson and Steven D. Hordley. Color constancy at a pixel. *J. Opt. Soc. Am. A*, 18(2):253–264, Feb 2001.
- [59] Graham D. Finlayson, Mark S. Drew, and Cheng Lu. Intrinsic images by entropy minimization. In *Proc. 8th European Conf. on Computer Vision, Prague*, pages 582–595, 2004.
- [60] J. M.A. Alvarez and A. M. Lopez. Road detection based on illuminant invariance. *Trans. Intell. Transport. Sys.*, 12(1):184–193, March 2011.
- [61] Graham D. Finlayson, Steven D. Hordley, Cheng Lu, and Mark S. Drew. On the removal of shadows from images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(1):59–68, 2006.
- [62] Jianbo Shi and C. Tomasi. Good features to track. In *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR '94., 1994 IEEE Computer Society Conference on*, pages 593 –600, jun 1994.

-
- [63] David G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110, November 2004.
- [64] Yan Ke and R. Sukthankar. Pca-sift: a more distinctive representation for local image descriptors. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 2, pages II–506 – II–513 Vol.2, june-2 july 2004.
- [65] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-up robust features (surf). *Comput. Vis. Image Underst.*, 110(3):346–359, June 2008.
- [66] Patrice Y. Simard, Leon Bottou, Patrick Haffner, and Yann LeCun. Boxlets: A fast convolution algorithm for signal processing and neural networks, 1999.