UNIVERSITÀ DEGLI STUDI DI PARMA

Dottorato di Ricerca in Tecnologie dell'Informazione

XXII Ciclo

DRIVER-VEHICLE COOPERATION. ANALYSIS AND EXAMPLES

Coordinatore:

Chiar.mo Prof. Carlo Morandi

Tutor:

Chiar.mo Prof. Alberto Broggi

Dottorando: *Pier Paolo Porta*

Gennaio 2010

Alla mia famiglia
e ai miei amici

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The research on intelligent vehicles and more generally on Intelligent Transportation Systems, is a very hot topic worldwide since all car makers, automotive suppliers, and many research centers are actively working in this field. A few Advanced Driving Assistance Systems (ADAS) are in fact already available on the market, such as Lane Departure Warning, Blind Spot Monitoring, radar based Automatic Cruise Control, and Collision Mitigation. However, all these systems are based on a limited perception of the world around the vehicle and their ability to intervene autonomously and alter the vehicle behavior is reduced to a minimum. State-of-the-art ADAS are in fact acting on braking only. Conversely, autonomous steering capabilities are currently very far from the market, with the only exception of the application of Electric Power Steering (EPS) to parking aids.

Car manufacturers and automotive suppliers are extremely interested in this research stream, but at the same time are very cautious in investing in this kind of long term and risky research. The benefits of such technologies are indeed clear to everyone, but they are committed to invest in short-to-medium term research that can lead to possible products in the near future: the development of autonomous vehicles, even if regarded as highly appealing, is currently considered as very risky due to the different technological and legal problems that still have to be faced. Great payoffs might be obtained once the research has proven that the technology is mature enough,

legal issues are solved, and users acceptance is confirmed.

Although the research on autonomous vehicles is not felt as a primary strategic area of investment in the short-to-medium term for the transportation industry, some research laboratories have been working on this topic for years and have developed a great deal of expertise which will be the basis for car makers' future investments.

Events like the Grand and Urban Challenges [1, 2] demonstrated the potential of autonomous vehicles; anyway such prototypes are not ready for a real road environment. In fact, the Urban Challenge took place in a controlled environment: a town closed to the public, only few vehicles driven by stunt men that carefully obeyed to traffic rules, total absence of pedestrians, traffic lights, or cycles; namely a friendly environment.

Indeed, the Urban Challenge is a huge step forward towards autonomous vehicles deployment, but its results are not readily applicable and are therefore very far from a possible market. Vehicles developed for the Urban Challenge can not drive in real traffic conditions since they have been designed for that specific event and can only work in that specific scenario. These considerations are also true for technologies developed in other intelligent vehicles projects, like the EU IP PReVENT [3] (whose demonstration took place in a closed circuit, in Versailles in 2007), or in other public demonstrations, like the *Millemiglia in Automatico* test [4] which, although taking place in real traffic conditions, was performed in a structured environment like highways.

Many of the most successful contenders in the Urban Challenge exploited expensive processing systems and sensors that, due to their cost, size, and need for a specific field of view, can not be installed on commercial vehicles. Some of these sensors were installed on the vehicle roof, behind the windshield, or in lateral positions leaning out the vehicle's shape, thus affecting the overall safety of the vehicle, driver's field of view, the drag coefficient, and, ultimately, the overall vehicle appearance. Computers, cables, and other adapters were generally filling vehicles' boots and often passengers or driver seats were removed to add more sensors or devices. Moreover, these equipments were conceived and installed for the race, namely for a restricted use, therefore they were neither automotive proof nor engineered for on

board installation. Nevertheless, the Urban Challenge has been of paramount importance and showed to the scientific community, the automotive sector, and the public that current technologies are almost ready for this application. Conversely, on the functional side, more sophisticated perception and planning systems are mandatory in order to cope with real and complex situations and to be able to precisely manage vehicle driving.

Public demonstrations, like the DARPA Urban Challenge or the PReVENT IP, showed that the control side has reached an advanced development level; in fact, some advanced control systems have already reached the market (ABS, ESP...). Most of the efforts of the Urban Challenge teams who already participated in the previous Grand Challenges were addressed to the development of highly improved perception and decision systems that had to deal with the higher complexity of a Urban environment rather than control modules. The control systems were derived from the ones already developed for the previous Grand Challenge, adapted for an urban environment, and indeed enhanced for an improved reliability. This is mainly due to the fact that the control problem is intrinsically more structured and better defined with respect to perception or decision problems. The development of a control system mainly requires dealing with vehicle kinematics, modeling, and limited sensing, and many research works already reached remarkable results. The issues still to be solved are mainly related to extreme conditions such as weather problems (ice, snow,...) or unfriendly environments (off-road driving, slippery or uneven terrain,...). Certainly, more work is needed to improve reliability; as an example X-by-wire systems have not yet reached sufficient robustness and redundancy levels to be installed on a series vehicle. Therefore, although additional work has to be performed on the control side, it is mainly required for engineering issues.

Conversely, developing effective and reliable perception and decision systems is definitely more challenging. The perception problem is very complex and even not well defined: a perception system must deal with a high number of different situations and complex scenarios. Objects in a scene have to be perceived and classified; a cluttered background –typical of a urban scenario and therefore of a common situation– can lead to misdetections and introduce a high level of noise in the results. Similar

considerations also apply to decision systems: to decide on the action to be taken, it is not sufficient to model the vehicle like it happens on the control side, but it is mandatory to model the real world as well; therefore, a huge number of situations must be taken into account. This shows why the development of perception and decision systems needs more research efforts than control tasks.

In this thesis the results of the research activity I have carried out during the three-year period spent as a PhD student at the Department of Information Engineering of the University of Parma are presented. My work focused mainly on intelligent vehicles and their applications, both ADAS and autonomous vehicles, in particular two examples of these topics will be presented in chapters 3 and 4.

An analysis on driver and vehicle interaction will be discussed in chapter 2, different levels of interaction are reflected in the level of automation present in each system. Finally in Chapter 5 a prototype vehicle to test the concepts described will be explained.

Chapter 2

The interaction between driver and vehicle

In this first chapter the interaction between driver and intelligent vehicles will be presented. Between manual driving and fully autonomous driving many levels are present, a detailed description of these steps will be given in order to place and describe current driver assistance systems (DAS) in the scheme. The two main research trends on intelligent vehicles are Advanced Driver Assistance Systems (ADAS hereinafter) and autonomous vehicles. The interest on autonomous vehicles has been high until the beginning of the new millennium, when ADAS systems started attracting the attention of the major automotive companies. In the last few years a new interest in autonomous vehicles grew up, mainly based on applications oriented to particular environments such as agricultural, mining, military, exploration.

ADAS works mainly in predefined environments such as highway, parking lots, queue driving, etc. the performance have to be high in their domain of applicability. Reduced cost and size are also mandatory to allow easy integration and thus a commercialization in automotive market.

On the other hand in autonomous vehicles development, cost and size of the systems are less important and their applicability is generally in protected or structured environments as cited above, but they have to deal with all weather conditions, all

	Basic DAS	*Advanced DAS*
Perception	Minimal and very focused	Complex
Automation	Short term and automatically triggered	Long term and triggered by the driver
Supervision	Not needed	Needed

Table 2.1: Basic and Advanced Driver Assistance Systems differences.

terrain and all scenarios.

Even if the final aim of the system is different, most of the underlying ideas are shared –such as for example perception, planning and actuation– but the different requirements in cost and integration lead to different sensing and processing engines. The main difference between the two branches is definitely the presence or the absence of the driver, this implies also a difference of responsibility in case of fault.

Considering only the basic DAS such as ABS, ESP, etc. it is possible to act without the possibility for the driver to override the system, because they feature limited sensing, straightforward decisions and actuations and are applicable in well defined scenarios; only a short term automation is involved. On the contrary, ADAS include complex sensing and actuation, the cost limitation does not allow to have a redundant sensing of the environment and their applicability is related to specific scenarios. Considering for example current ADAS, their technology might be sufficient to drive without human intervention, at least in some situations such as for example highway environment. A vehicle equipped with Automatic Cruise Control (ACC) and Lane Keeping Assistant (LKA), is able to manage lateral and longitudinal control and thus drive autonomously in a highway environment. However in current legislation is not clear the responsibility in case of accident during autonomous driving thus driver supervision is needed since they are designed to complement the driver not replace it.

Driver supervision is needed when incorrect behaviors are possible: if the system

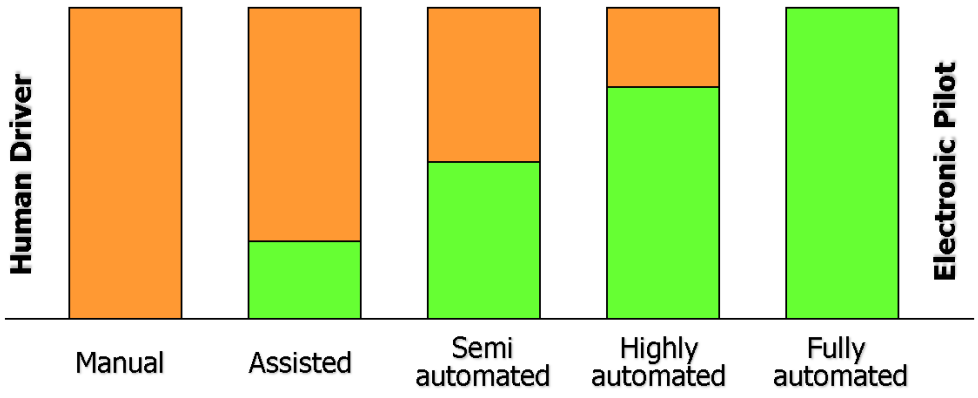


Figure 2.1: Taxonomy for vehicle automation.

is fault-aware, it informs the driver and fall-back to manual driving; in other cases the driver needs to override the automatic function to regain the control of the vehicle: for this reason engage and supervision of basic and advanced DAS are different as shown in the table 2.1.

In particular, while in a short term automation –typical of basic DAS– the system quickly recovers from a dangerous situation and then switches off, during a long term automation relieving the driver of the driving action can result in a lack of attention ad cause a potentially dangerous situation when the automation ends and the system fall-back to manual, because the driver attention at that time is low.

To avoid this problem, current ADAS want the driver to prove to be officially in charge of the driving action: the system performs a periodical check on driver's attention to keep it high, for example checking that he is holding the steering wheel.

2.1 Taxonomy for vehicle automation

Different levels of automation are possible between manual driving and fully autonomous driving. In particular five steps have been highlighted

- Manual driving

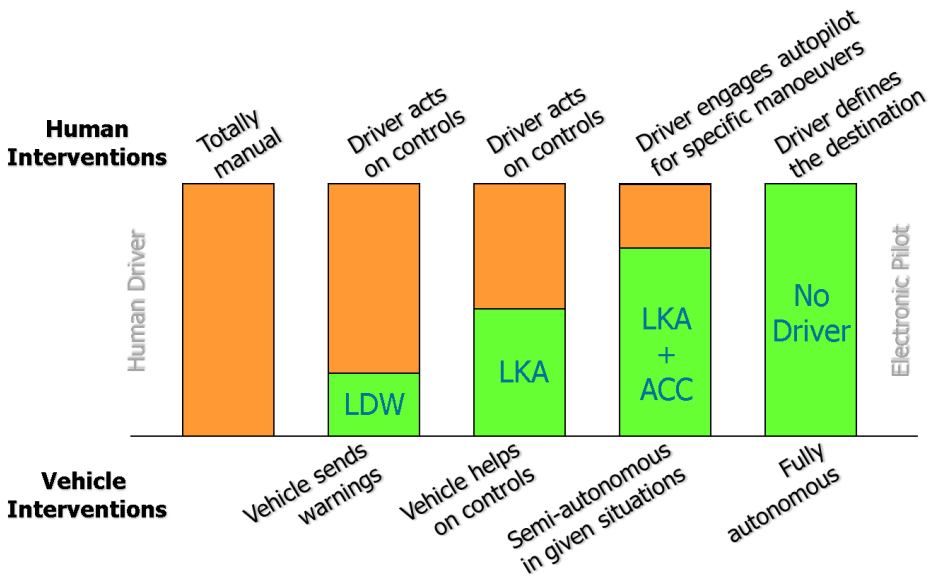


Figure 2.2: Car-driver interaction

- Assisted driving
- Semi automated driving
- Highly automated driving
- Fully automated driving

as shown in figure 2.1.

The first step after manual driving consists in assisted driving, where the vehicle only issues suggestion to the driver. In this level are included system like Lane Departure Warning (LDW), with no actuation on vehicle control but only an acoustical or haptic feedback to the driver is given.

Semi automated level is characterized by selected automation that helps the driver in some situation they involve only a short intervention on actuators to correct and recover from potentially dangerous situations. A Lane Keeping Assistant (LKA) is part of this level, the driver still acts on vehicle control, but sometimes he is helped

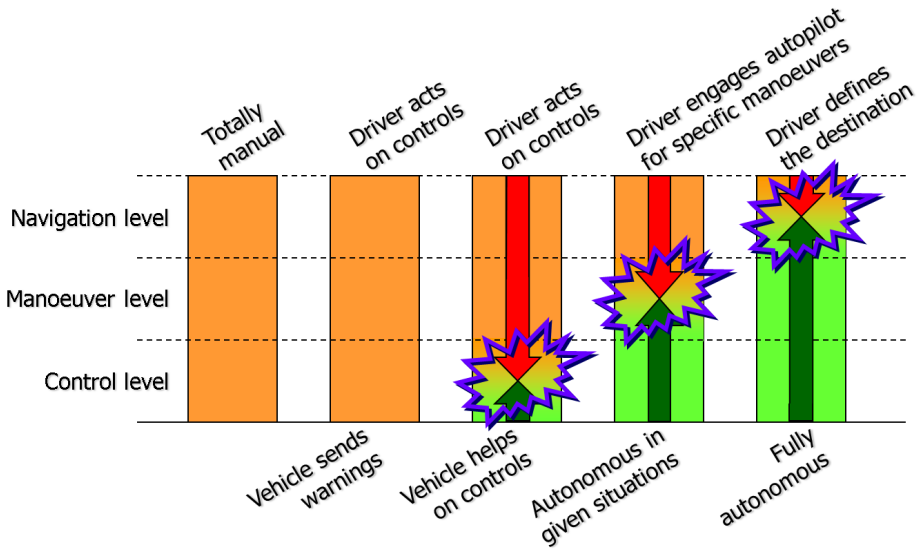


Figure 2.3: Conflicts highlights in car-driver cooperation.

by the intelligent vehicle. Automatic Cruise Control (ACC) belongs to this level too: the car keeps a safe distance from the preceding vehicle when the system is engaged.

In highly automated level the vehicle can make maneuvers and holds the control even for long time in specific scenarios. In this case for example, the car and the driver meet at maneuver level. An example of this level can be LKA+ACC: the vehicle is able to keep a safe speed and a correct distance to the vehicle ahead, and follows the road. It involves lateral and longitudinal control with full autonomy in a specific and limited scenario, highway environment in this case.

The last step is the fully autonomous driving. The driver have to define only the destination and all the driving action, including planning, is demanded to the intelligent vehicle.

In high levels the vehicle proposes not only suggestions or corrections, typical behavior of short term automation, but solutions to the driver via various feedbacks. Due to these different interaction levels, even different HMIs have to be employed as described in section 2.2. The higher is the level of automation the higher is the level

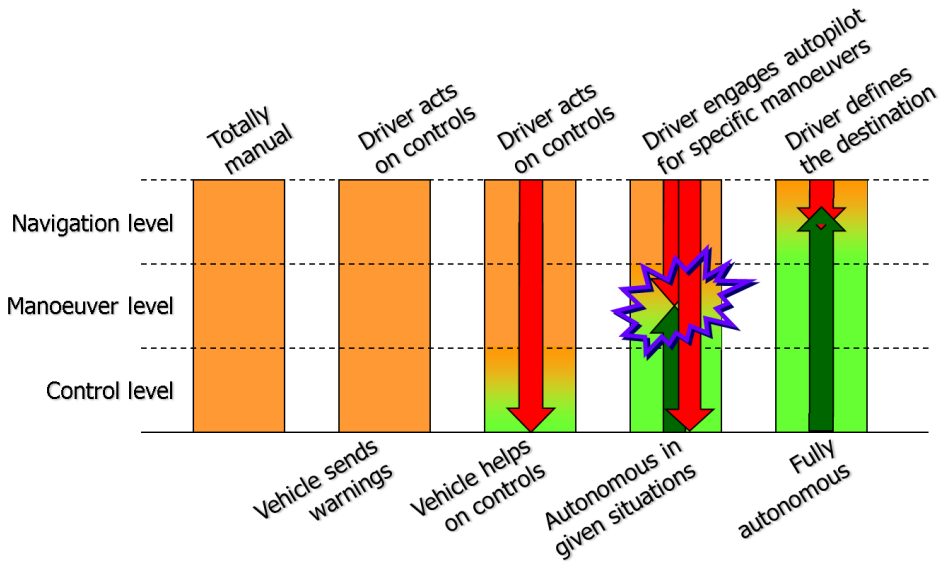


Figure 2.4: Conflicts are resolved on different levels. The driver can intervene in a lower level to resolve the conflict.

of interaction in car-driver relationship as shown in figure 2.2.

2.2 Conflicts resolution

Except for the first and the second level, conflicts are possible and an arbitration is needed to solve these situations.

The conflicts can be related to different approach at the same task or to an opposite interpretation of the perception, or even to a misunderstanding, or to a fault. Conflicts may happen at different levels: navigation level, maneuver level, and control level as shown in figure 2.3.

In the semi automated level, where short term automations can be engaged by the vehicle, conflicts between driver and vehicle are at a low level, the control level. The intelligent vehicle intervenes when detects that unsafe behaviors of the driver or dangerous situations occurs. In this case conflicts can be solved giving the driver the

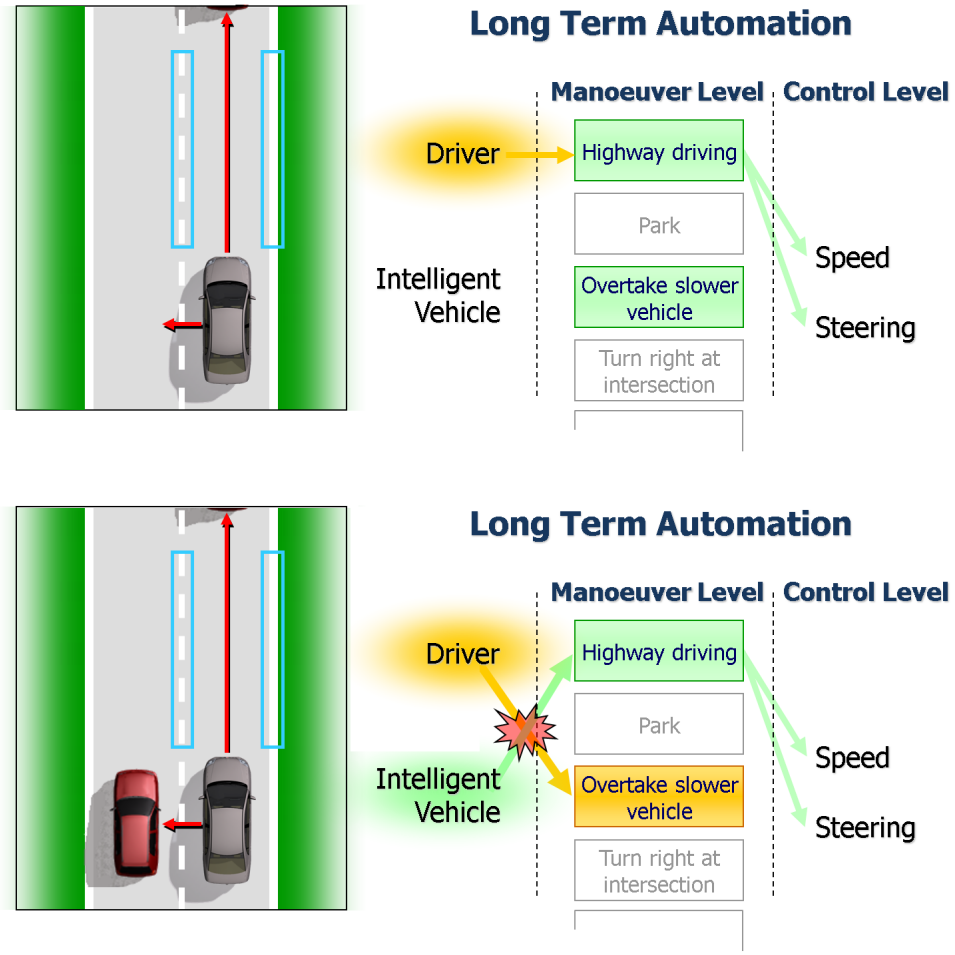


Figure 2.5: Car-driver cooperation in long term automation. A possible conflict is shown. The driver issues a maneuver that the vehicle judges as unsafe.

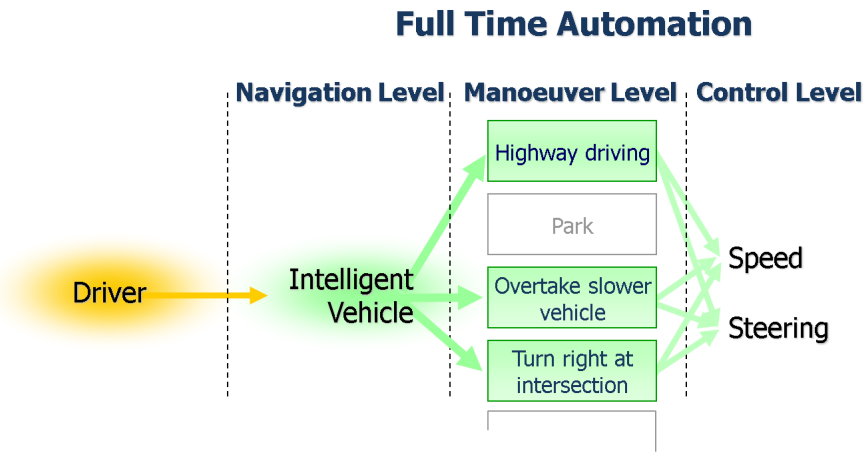


Figure 2.6: Car-driver cooperation in full automation.

possibility to override the vehicle.

In the highly automated level the interaction between vehicle and driver is different. Let's consider the following example: during highway driving, first the driver is holding the complete control of the vehicle; if, for example, the intelligent vehicle locates the lane markings and the vehicle ahead, a set of possible maneuvers can be proposed to the driver (i.e. highway driving, overtake slower vehicles, etc.). The driver can select one of them and start a long term automation engaging the autopilot, this interaction lies at an higher level than before. The driver can also intervene at any time selecting a different maneuver; on the other hand the system may judge the maneuver issued as potentially unsafe and refuse to follow the command. In this case a conflict is present at maneuver level.

An arbitration is required: there is no possibility to distinguish between unsafe driver or vehicle fault, both cases are possible and there isn't enough information to discriminate between them. The conflict can be solved acting on different levels: at maneuver level the vehicle will not follow commands believed as unsafe, but the driver holds full overridability on the system at lower level (control level). With this arbitration the driver can regain control of the vehicle by intentionally intervening at

control level.

Different HMIs are thus mandatory for this conflict solving implementation. A touch screen can be suitable to manage the maneuver level, while gas and brake pedals and steering wheel can be the interface for the control level

For the last level, full time automation, the car-driver interaction occurs at navigation level. The driver specifies the destination and the route planner helps the vehicle selecting the sequence of correct maneuvers. Even in this case, the driver will use different HMIs to intervene at different levels and resolve conflicts: during autonomous driving is possible to issue maneuvers such as turn right at the intersection forcing a replan of the route, or take control of the car at lower level.

In conclusion, for ground vehicle, conflicts are solved in different ways depending on the level of occurrence. Figure 2.4 summarizes the conflicts resolution on the different levels.

In the next chapters an overview of two examples of different level of automation will be given. Chapter 3 describes an ADAS example, while chapter 4 presents an example of autonomous vehicle, focusing on a particular sensing capability.

Chapter 3

An example of ADAS: traffic sign recognition system

In this chapter an example of ADAS will be given. In particular the application described can be considered part of the *Assisted* driving described in section 2.1 since no actuation is currently available. Nevertheless since the perception phase involves complex sensing, this is considered an ADAS.

The approach developed will be described in details both in detection and classification stage in the following sections.

3.1 The system

Automatic traffic signs detection and classification is a very important issue for Advanced Driver Assistance Systems (ADAS) and road safety: different road signs detectors were developed in the last 10 years [5]. Most of the industrial systems developed are based only on speed limit signs recognition, on the contrary the system proposed here can detect a large scope of road signs. The system described here [6] works with a camera already mounted on-board for other purpose such as lane departure warning (LDW). Another advanced feature introduced here is the low dependence from illumination conditions: this is of paramount importance for good

performance in early mornings and late afternoons where sunlight usually presents an appreciable deviation towards red.

Missed signs can cause dangerous situations or even accidents. An automatic road sign detection system can be used both to warn drivers in these situations, and to supply additional environmental information to other on-board systems such as ACC[1].

Both gray-scale and color cameras can be used for this purpose; in the first case the search is mainly based on shape and can be quite expensive in terms of computational time [7, 8]. Using a color camera, the search can be based mainly on color: color segmentation is faster than a shape detection, although requiring additional filtering. Images acquired by an inexpensive color camera can suffer from Bayer conversion artifacts and other problems such as color balance, but anyway, the developed system is more robust and definitely faster. Some research groups have already used color images for traffic signs detection. Most of these methods have been developed using color base transformations; HSV/HSI color space is the most used [9, 10] but other color spaces, such as CIECAM97 [11], can be used as well. These spaces are used because chromatic information can be easily separated from the lighting information: this is used to detect a specified color in almost all light conditions. Anyway traffic signs can be detected in RGB [12] or YUV [13] color space with the advantage that no transformation, or just a very simple one, is required. The segmentation and thresholding algorithms are anyway more complex, but a lot of computational time can be saved. In order to make the detection more robust, both color segmentation and shape recognition can be used in cooperation [11]. Anyway the processing described so far has to be computationally light to keep the advantages of the selected color space.

Many different approaches are used for the subsequent classification: most of them are based on artificial intelligence techniques; the most used are neural networks [14, 15], bayesian networks, and fuzzy logic [16].

The proposed approach [6] is based on 3 steps (see figure 3.1): color segmentation, that is presented in the following section, shape detection (section 3.3) and

[1] Automatic Cruise Control

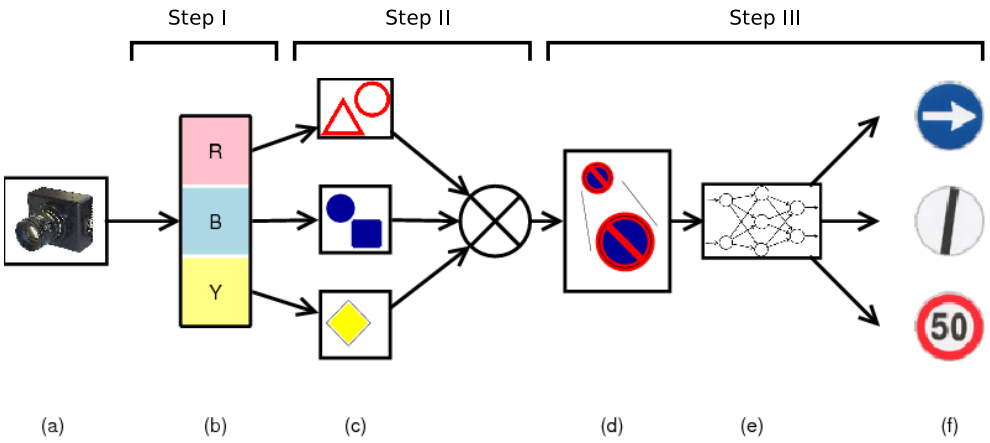


Figure 3.1: Algorithm flowchart. (a) Image acquisition; (b) color segmentation; (c) shape detection; (d) bounding box scaling; (e) classification; (f) output.

classification, (section 3.5) based on several neural networks.

3.2 Color analysis

In this section the first step of road signs detection based on color information is presented. One of the main road sign feature is their immediate identification by a human driver: this is due to a limited number and very specific set of shapes and colors. In particular, the colors set used for road signs is composed of: white, grey, red, yellow, and blue; green used on highway signs (in Italy) is not considered in this analysis.

This section presents a discussion on the choice of the color space to use, a robust color segmentation, and a solution for the problem of chromatic predominance of the light source.

3.2.1 Color segmentation

The specific illumination condition deeply affects the road signs color perception. Common environmental conditions are usually characterized by a wide range of dif-

Figure 3.2: Examples of different illumination conditions.

ferent illuminations: direct sunlight, reflected sunlight, shadows, and sometimes even different illuminations can coexist on the surface of the same sign as shown in figure 3.2.

The objective of this work is to identify a sign of a given color (for example red) regardless of its illumination. As already mentioned, in literature most approaches are based on HSV or HLS color space [10, 16, 13] but the camera used for this application has a raw Bayer output and a conversion would be too computationally expensive because of the non-linearity introduced. Therefore RGB space is too dependent on brightness, so different color bases were tested to find one that is less dependent on brightness. All the color base conversions evaluated can be obtained with linear transformations; this kind of approach has been followed also by [12, 17, 14, 13, 18].

First YUV color space has been tested, using the matrix shown below:

$$\begin{pmatrix} Y \\ U \\ V \end{pmatrix} = \begin{pmatrix} 0.299 & 0.587 & 0.114 \\ -0.147 & -0.289 & 0.436 \\ 0.615 & -0.515 & -0.100 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix} \tag{3.1}$$

The Y coordinate is strongly dependent on brightness thus, considering only the UV plane, it is possible to bound regions mainly based on hue. Figure 3.3 shows the UV plane with Y=127: the grey color is visible in the center of the image.

Empiric tests demonstrated that this kind of bounding is too simple to collect all the cases of different illumination and to cover most of the case study.

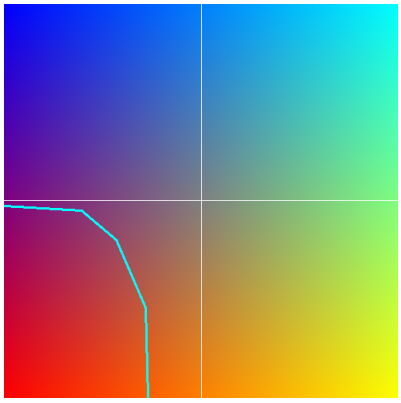


Figure 3.3: The UV plane in the YUV color space with Y=127. The blue line bounds the region selected to identify the red color.

The second attempt focused on RGB values: what we bound here is the ratio between different channels and not only the channel itself. Eq. 3.2 shows the expression of this kind of thresholding:

$$\begin{cases} \alpha_{min} * G < R < \alpha_{max} * G \\ \beta_{min} * B < R < \beta_{max} * B \\ \gamma_{min} * B < G < \gamma_{max} * B \end{cases} \tag{3.2}$$

The values of the parameters involved have been obtained tuning the algorithm on real images in different conditions.

Both these methods have produced good results but the second one is chosen for several reason:

- lower number of false positives,
- lower computational load: indeed no color space conversion only a thresholding is needed,
- easier to tune: because it is less sensible to small parameters variations.

After an analysis on colors present in Italian traffic signs the research has been focused on 3 colors: red, blue, and yellow; the described algorithm is applied on all

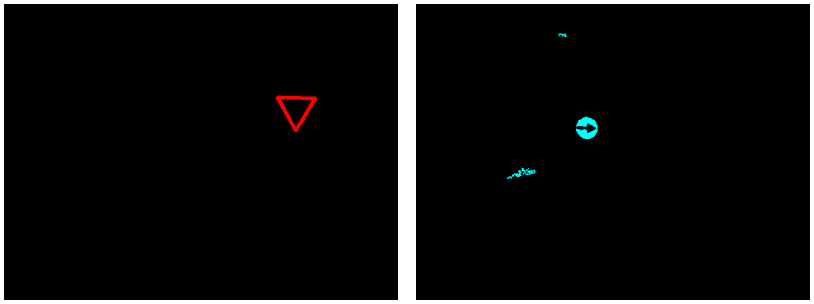


Figure 3.4: Color segmentation of red and blue channel.

these 3 colors to generate 3 binary images containing only the pixels referred to that color; figure 3.4 shows the results for red and blue.

All the pixels referred to a same connected region are labeled together. Regions smaller than a fixed threshold are discarded because usually do not represent a road sign or, otherwise, the target is so far that a detection would not be followed by a successful classification due to reduced size.

Considering for example the red color segmentation, all the red objects present will be detected (e.g. cars, buildings, placards), but these detections will be discarded by the following further steps.

3.2.2 Chromatic equalization

One of the main problems experienced in this stage is the dependence on the color of the light source. For example during sunset or dawn a red color predominance is present and this deeply affects the color segmentation step; see figure 3.5.

To solve this problem we developed a chromatic equalization based on two steps:

- light source color identification,
- chromatic correction.

The easiest way to find the light source color is to find an object supposed to be white and then compute the aberration from theoretical white (255, 255, 255 in

Figure 3.5: Original image and result of chromatic equalization. In the right image the region supposed to frame the pave is shown.

the RGB color space). Unfortunately on a dynamic environment such as roads, it is difficult to have a white reference point. Thus the color of the road is searched for, as suggested in [19] that is supposed to be grey. In [19] the chromatic response of several materials has been analyzed in different illumination conditions. Moreover in most cases in our vehicular application a specific region of the image frames the pave as shown in figure 3.5.

Through the use of a temporal window in which we integrate the light source color, it is possible to avoid fast changing of the result and keep it stable. In case of a tracking has to be introduced in the processing chain, it is very important to have stable conditions for a reasonable number of frames.

Once the light source color has been evaluated, chromatic equalization can be applied. This step is very similar to a gamma-correction process: to reduce the computational time a linearization of the gamma correction function has been used, as shown in figure 3.6 and described below.

line A: $y = \alpha \cdot x$

line B: $y = \beta \cdot x + \gamma$

line C: $y = -x + k$

D: point of coordinates (255, 255).

Now suitable values for the parameters α, β, γ and k have to be found. The α value for the 3 channels can be computed in 3 steps:

- consider the RGB value of the light source color,
- set $\alpha = 1$ to the channel that has the intermediate value,
- set $\alpha = \frac{intermediate\ channel\ value}{channel\ considered\ value}$ for the other 2 channels.

The parameter k can be set once with empiric tests, to avoid saturations and β can be obtained forcing the curve to be continue and to reach D.

3.3 Shape detection

After color segmentation, a first sorting based on shape is performed. This sorting is developed in order to reduce the complexity of the final classification. Two different methods are used to determine the correct shape with high reliability; the first one is based on pattern matching, the other one is based on remarks about edges. Before this sorting, a method to merge and split the bounding boxes generated by color segmentation is applied together with a filtering based on aspect ratio.

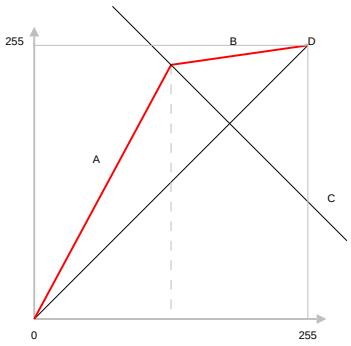


Figure 3.6: Gamma correction curve.

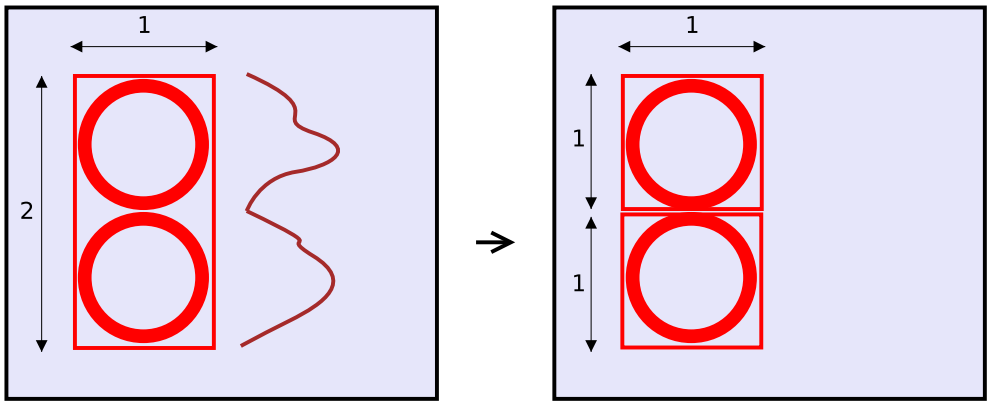


Figure 3.7: Bounding boxes split.

Another method for shape detection based on Sobel phase analysis will be described as well. This approach is more robust to sign rotation but is not suitable for circle shape detection.

3.3.1 Bounding Boxes merge and split

Color segmentation can sometimes provide a bounding box that contains two or more signs or only a part of a sign; the use of such a bounding box in the subsequent classification may cause an error: this merge and split step is developed in order to solve this problem.

When two signs with the same color are hanged on a single pole it may happen that the segmentation identifies the two signs as a single one because of the weak separation between the signs. All the boxes with height almost double as width are checked considering the vertical histogram of the binarized image. If a very low value is identified around the middle of the histogram, the bounding box is splitted, see fig. 3.7.

On the other hand, bounding boxes of the same color, that overlap more than a threshold, are merged together as a single sign. This process is useful to merge in a single box different parts of a same sign that may have been divided by the labeling

step.

A single sign can also contain two colors that can be identified in the two corresponding images, e.g. work in progress signs contain both red and yellow. All the three color segmentation images are checked for overlapping bounding boxes and, if overlapping bounding boxes are found, they are merged in a single bounding box. Each bounding box, can handle up to two colors, a primary and a secondary one; if a bounding box is the result of a merging, its primary color will be set to the color of the larger bounding box while the other bounding box will set the secondary color. If a bounding box does not overlap any other box, it will have only a primary color.

3.3.2 Pattern matching

Bounding boxes with primary color red or yellow are supposed to have specific shapes that can be easily detected using a pattern matching. A reference pattern is built for each shape that have to be detected. The patterns are built growing the shape of a signs, with the aim of detecting also rotated or misaligned signs; these patterns are shown in figure 3.8. Triangle, reversed triangle, circle and filled circle are searched in red bounding boxes, while only rhombus is searched in yellow ones. Filled circles are used to detect stops and no thoroughfare signs.

All the detected bounding boxes are resampled to a fixed size (50×50 pixels) equal to the pattern size. A very simple pattern matching is used in order to reduce the complexity and the computational time: the matching function counts the number of

(a) (b) (c) (d) (e)

Figure 3.8: Patterns of searched shapes: (a) circle, (b) filled circle, (c) triangle, (d) reversed triangle, (e) rhombus.

pixels that are positive both in the pattern, that is a binary image, and in the image of the corresponding color. The ratio of matched pixels to white pixels in the considered pattern is computed. When all the patterns have been checked, the shape that provides the best ratio is chosen as the correct shape of the sign, only if the ratio exceeds a fixed threshold. If the chosen shape is a circle, the ratio of matched pixels to white pixels in the image of the corresponding color is considered as well, in order to distinguish between circles and filled circles.

3.3.3 Remarks about edges

A double check is extremely useful to be sure that the chosen shape is the correct one. Starting from the center of each bounding box in the segmented image and expanding to the borders in several directions, the last pixel of primary color is searched for in order to find the external edges of the labeled region inside the bounding box. The set of points found is used to generate lines or a circle that best fits the set. Appropriate geometrical formulas are used to determine if the lines are composing one of the searched shapes. The last step is to find the best match between all this possible cases.

The result of this processing is used to dynamically modify the threshold of the pattern matching previously described: if the two methods agree the pattern matching threshold is reduced, otherwise it is raised.

Since blue signs are not suitable to be detected with pattern matching because their shape fills most of the bounding box, an ad-hoc method has been developed. It consists in searching for the blue pixel nearest to each corner. According to the distances found we can discriminate between squares, circles and other irregular shapes.

3.4 Shape detection based on Sobel phase analysis

Another method for shape detection has been investigated, based on the evaluation of the Sobel edges and Hough images in a region of interest detected by the color-based stage. The system proposed is a simple shape detector independent from geometric distortion i.e. rotation, partial occlusion, deformation, translation.

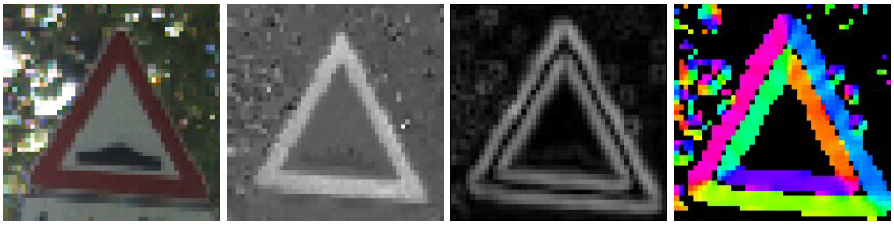
Many approaches are presented in literature for shape detection: [12, 20, 21] show that pattern matching is a robust and fast method; in [22] Hough transformation based on radial symmetry is used to detect speed signs. Genetic algorithms, used in [10, 23, 24, 25], allow accurate results in shapes detection, but their execution requires computational times unsuitable for real-time applications. Other methods ([26, 27, 28]) are based on the use of supervised learning methods for classification, like SVM. Some methods recognize only a specific shape: [17, 14, 16] present a triangular shapes detector; in [22, 12] only circular shapes are detected; the algorithm presented in [29] allows triangles, rectangles and octagons detection, through the use of a fast radial symmetry and the shape center detection. In [6] has been presented a vertical traffic sign recognition system based on a three-step algorithm: color segmentation, shape recognition and neural network classification to detect and classify almost all Italian traffic signs. The shape detection method described was sensible to translation and rotation of signs. To address this problem the proposed approach detects edges on images, correlating them with their gradient distribution: the shape of a placed object depends on the edges and on their features i.e. position and mutual orientation.

3.4.1 Edges detection

A conversion from RGB to grey scale images is performed to detect edges: each region limited by a bounding box is cropped from the color image (figure 3.9a) and converted to a grey scale crop (figure 3.9b). Since the RGB to grey conversion causes information loss due to the passage from a vector field (color image) to a scalar field (grey image), in order to increase the $\frac{signal}{noise}$ ratio, the predominant color component, according to the primary color of each bounding box detected, is extracted from original crop and pointed out in that grey. The following formulas are used for this conversion.

$$Red : min\left(\frac{r}{max(g,b)} * 85\,,\, 255\right)$$

$$Blue : min\left(\frac{b}{max(r,g)} * 85\,,\, 255\right)$$



(a) (b) (c) (d)

Figure 3.9: Shape detection based on Sobel phase analysis processing steps: (a) Crop of the sign to be detected. (b) Grey scale conversion with primary component highlighted (in this case red). (c) Sobel norm image. (d) Sobel phase image.

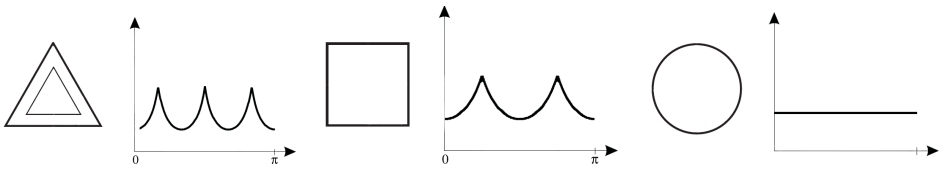
$$Yellow : min\left(\frac{min(r,g)}{b} * 85\,,\,255\right)$$

The factor 85 has been experimentally computed to best enhance the primary color contribution.

In order to enhance edges, Sobel operator with a 5×5 mask is applied for each grey scale crop, obtaining images about the Sobel norm (figure 3.9b) and the Sobel phase (figure 3.9c) intended as $arctg(G_y/G_x)$.

3.4.2 Analysis of the Sobel phase distribution

Using images obtained by Sobel filtering, analysis about the phase distribution of Sobel edges is performed to detect distinctive features for each shape. This step focus on the study of the most frequently edge gradient for each region of interest detected, to decide if the placed object has a road sign compatible shape and which one is. The main purpose of this step is the study of the peaks in the sobel edges phase distribution, regardless the sign of the transition described by each edges. Since opposite transitions cause in the Sobel phases image angles at a distance equal to π, statical results represented in the range $(-\pi, \pi)$ are mapped to the range $(0, \pi)$, because at this step we are interested in the study of the edge gradient most frequent directions.



(a) (b) (c)

Figure 3.10: Histogram of the ideal Sobel phase angles distribution, for the outlines of (a) concentric triangles, (b) rectangles, and (c) circles.

Figure 3.10 shows, for each considered shape, the ideal trend of its phase distribution; according to the shape of interest the distinctive features are:

(a) 3 peaks placed at a distance equal to $\frac{\pi}{3}$ for triangles;

(b) 2 peaks set at a distance equal to $\frac{\pi}{2}$ for rectangles;

(c) histogram with all equal values for each considered angle in case of circles.

Some examples relative to phase distribution of real signs are reported in figure 3.11: empirical results show that the phase distribution of danger signs (figures 3.11a, 3.11b) and indication signs (figures 3.11c, 3.11d) substantially follow a common trend similar to the expected ones shown respectively in figure 3.10a and figure 3.10b. Otherwise the histograms of figures 3.11e, 3.11f show as the phase distribution of two different types of circular signs are different from each other and far from the ideal trend of figure 3.10a. This is due by the symbols drawn on the sign and by the elements placed on the background that introduce significant oscillations on the phase histogram involving a trend different from the ideal constant one. The same noisy elements affect also the phase distribution of triangular and rectangular shapes, but in these cases the distinctive peaks of the phase histogram are greater than the oscillations introduced by the noisy components.

Statistical analysis on the real signs is performed in order to evaluate, as distinctive feature, the real gap among peaks in the Sobel phases distribution according to the shape delimited by the bounding box. The distance between two peaks is estimated as the minimum of circular distance, calculated in clockwise and anticlockwise

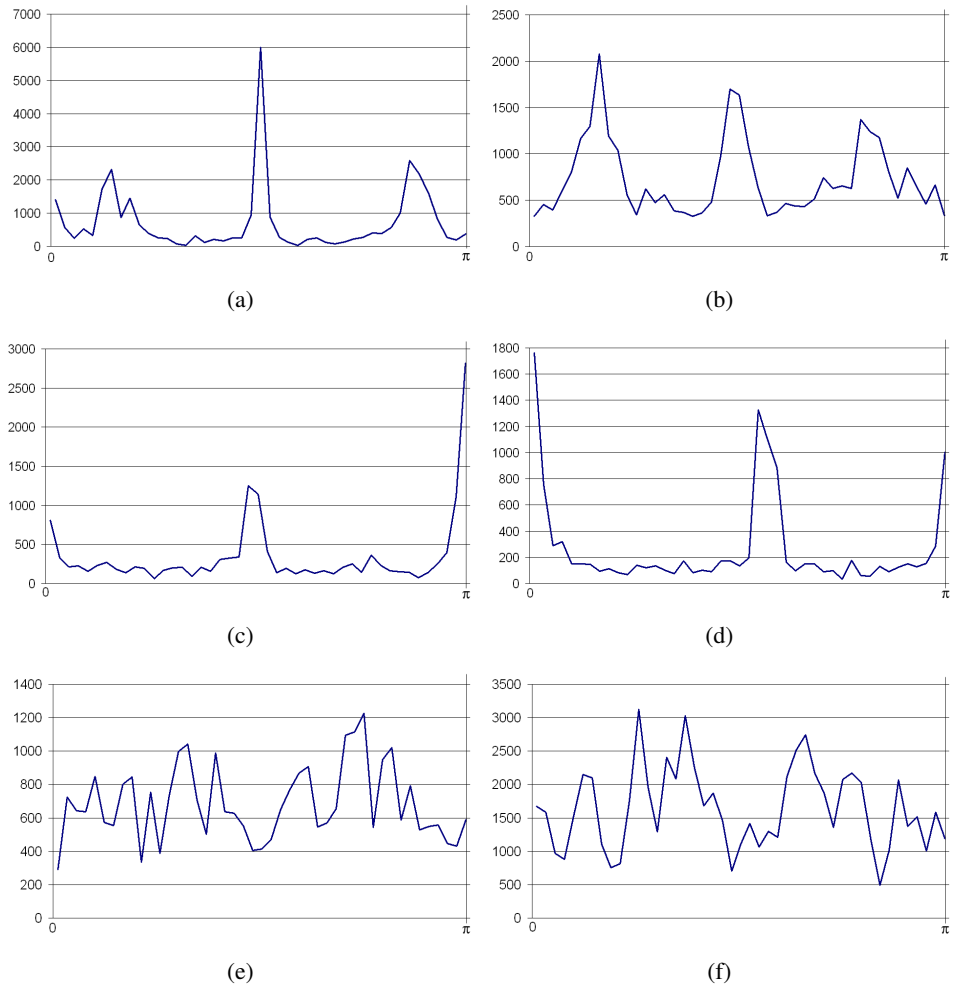


Figure 3.11: Histograms of the Sobel angle phase distributions for: (a-b) triangular signs, (c-d) rectangular signs, and (e-f) circular signs. For each considered phase angle (in the range $(0, \pi)$), we calculate the number of pixel having that phase or its supplementary.

directions:

$$min\left(\, abs\big(v1-v2\big)\,,\, range-\Big(abs\big(v1-v2\big)\Big)\,\right)$$

Since the minimum circular distance between two peaks in the range $(0,\pi)$ could vary between 0 and $\frac{\pi}{2}$, for the analysis of the real gap we consider discrete intervals of the range $(0,\frac{\pi}{2})$. i.e. the x-axis is split in 23 discrete intervals of width equal to $4\frac{\pi}{180}$ (4 degrees). On figure 3.12 are shown the bar charts of the obtained results: x-axis represent all possible gaps among the phase distribution peaks, centering the peak on each discrete range considered; while the ordinate shows in logarithmic scale the number of all bounding boxes characterized by each gap of the interval $(0,\frac{\pi}{2})$. The first three intervals (i.e. gap less than 10 degrees) are discarded because, according to the implemented method for the peaks searching, after the detection of the first high value, its nearly points are not considered during the searching of the second peak, in order to avoid the detection of a queue of the first one. This involve that the minimum distance measured between two peaks is at least 10 degrees.

According to the expectations, as shown in figure 3.12, bounding boxes with triangular shape are concentrated in a neighborhood of the ideal peak gap value i.e. 60 degrees: for the selection of the largest number of regions looking for triangles, we consider as possible candidates all bounding boxes having a peak gap in a range on the left and on the right of ideal peak gap value. The range value has been experimentally computed to be 12° on each side. Bounding boxes with rectangular shape present a peak gap value close to the last gap sample considered equal to $\frac{\pi}{2}$. The threshold to forward a bounding box to the rectangles processing is in the neighborhood equal to the three last represented intervals.

3.4.3 Triangles and rectangles detection

Triangles

All triangular road signs (danger, work in progress and yield) have a thick red border that generates two concentric triangles (figure 3.13): the biggest coincide with the edges that separate the sign from the background, otherwise, the smallest delimits

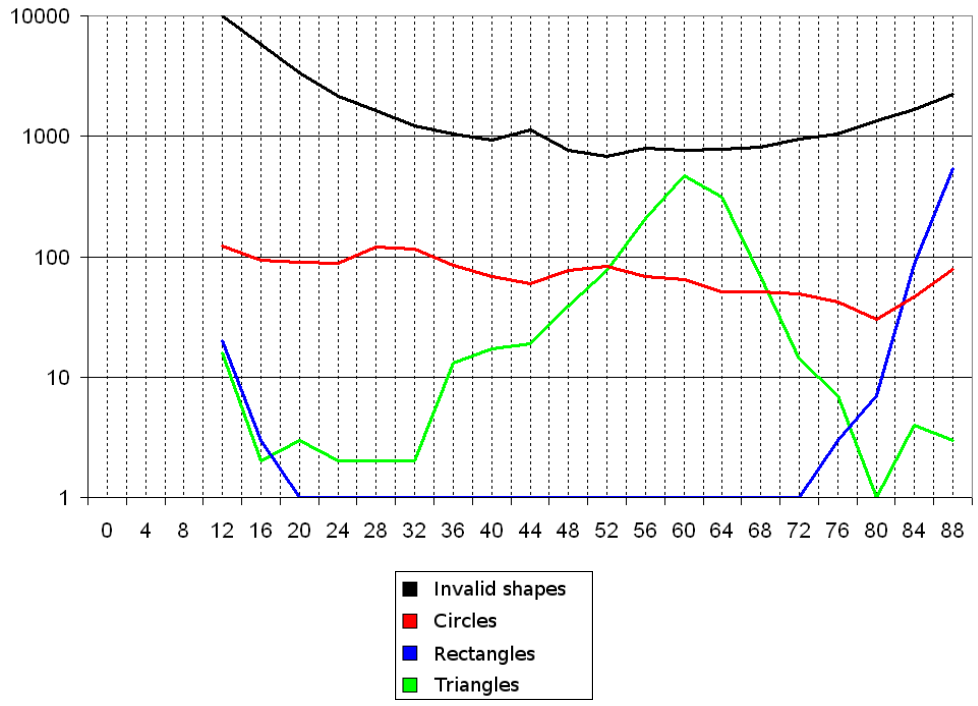


Figure 3.12: Chart showing, according to the shape, the number of bounding boxes for each gap value in the range $(0, \frac{\pi}{2})$.

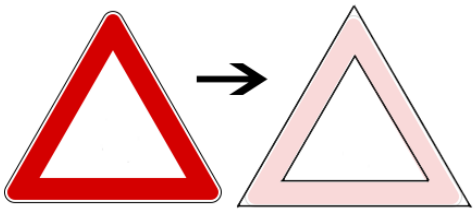


Figure 3.13: Concentric triangles in a danger sign.

Figure 3.14: Example of sign where the edges of the inner triangle are not well contrasted; on the other hand the border separating the sign from the background is visible.

the inner area (white or yellow) where symbols are painted.

The aim of this step is to verify that candidates selected by the peak gap analysis of the Sobel phase distribution match a triangular shape, and then crop the inner area that frames information for the classification stage. We are also interested in the detection of the edges separating the sign from the background because they could contribute to the detection of the inner triangle when the bounds of the inner area are not well contrasted (figure 3.14).

To reach this goal we detect in the candidate crops all lines that could generate a two concentric triangles compatible with the presence of a triangular road sign: the mutual position of this lines will allow to verify the presence of a triangle, defining its orientation and validating the relative bounding box. A valid bounding box could be then set with more precision, in order to crop only the inner part of the sign to

(a) (b) (c) (d)

Figure 3.15: (a) Absolute Sobel values image. (b) Sobel phases image represented in chromatic scale. (c) Categorization. (d) Labeling and blob detection.

be processed by the neural network. To discriminate between the inner and the outer edge of the red strip we need to consider the sense of the gradient. The searching of these edges is performed independently from their position. To detect the six lines that generate the six sides relative to the concentric triangles, we consider six different membership categories for the pixels of the image (seven counting the pixels that do not belong to any categories), according to the value of the Sobel phase (i.e. direction and sense of the gradient). This categories are associated to the six principal directions of the lines to be detected: three for the outer triangle and three for the inner one. The algorithm calculates six ideal angles, compatible with the presence of two triangular concentric shapes; these angles are fixed at multiple distances of $\frac{\pi}{3}$, starting from the first peak angle detected in the phase distribution. We consider a neighborhood of $\frac{\pi}{3}$ centered on each angle, in order to recover the domain $(0, 2\pi)$; every range will define a different category. According to the value of the Sobel phase, every pixel is classify as member of a category. After the categorization we obtain a grey scale image, where pixels could assumed only seven values (six categories and the value 0 to distinguish all points to be discriminated). An example of categorization is shown on figure 3.15c.

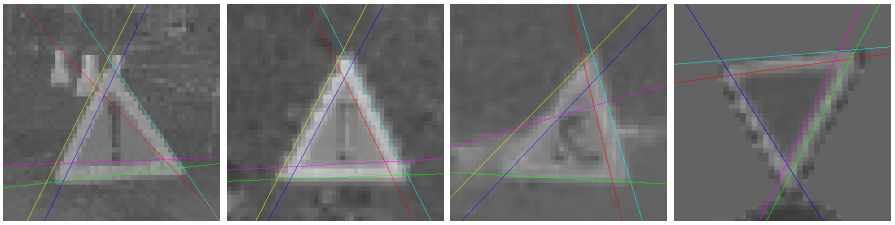
To reduce the noise in the image, a labeling procedure is performed, isolating connected components and eliminating the small ones according to a threshold related to the bounding box dimension (figure 3.15d). The six lines delimiting the two

(a) (b)

Figure 3.16: (a) Label image with split blobs. (b) Application of the Hough transform generating a line for each blob detected.

concentric triangles are detected using Hough transform. We found that generating a line for each blob detected may introduce noise: two or more blobs, referred to the same category, could be split (figure 3.16a), because of the presence of discontinuities or partial occlusion of the road sign; this separated blobs involve the generation of similar lines that affect the shape detection (figure 3.16b). To avoid this Hough transform is applied on the Sobel norm image, processing distinctly points of different categories and considering at the same time points belonging to blobs relative to the same category. Six different Hough images are computed, for each considered category, only evaluating nonzero pixels of the Sobel norm image, belonging to a valid blob. In order to obtain more accurate results, reducing the noise contributions, we also calculate a weighted Hough transform, evaluated considering the intensity of the edge variation described by every pixel of the Sobel norm image. Considering the maximum values of the Hough transforms, the algorithm detects at most six lines (figures 3.17a, 3.16b, 3.16c): the presence of slight edges could causes the absence of blobs for a category involving the generation of less than six lines (figure 3.17d).

In order to crop the area to be processed by the neural network, the vertexes generated by the intersections of the lines delimiting the inner triangle are detected. If the lines obtained by the Hough transformation are six, we search the three pairs of them delimiting the thick red edge of the sign; i.e. pairs of ideal parallel lines, with distance between phases equal to π are searched. Otherwise when the number of lines obtained is less than six, at least three lines with three different orientation



(a) (b) (c) (d)

Figure 3.17: Example of detections. The number of lines detected is less than six when there are no valid points for a given category.

are detected in order to obtain through their intersection a pattern of sides compatible with a triangle: in figure 3.18 are shown all valid configurations for the generation of triangular shapes. Configurations (c) and (f) in the figure 3.18 are formed by sides of the same type (i.e. they are all sides of the inner triangle or all edges of the outer triangle). The other combinations are generated by two sides of same the triangle (inner or outer) and one of a different type (outer or inner).

Basing on this considerations, we are able to find what type of pattern the three detected lines form: the position of their intersection points defines the orientation of the triangle while the information about mutual gap between the peaks of the phase distribution and the orientation, allow to recognize the type of side (i.e. inner or outer). Then we consider the intersections between the detected lines in order to find the three vertexes delimiting the inner triangle: if one of the three detected sides is referred to the inner triangle, while the other two belong to the outer triangle (figure 3.19a), two vertexes (i.e. those generated respectively by the intersection between each outer side and the inner side) are shifted along the line relative to the inner edge, while the third point (i.e. the intersection between the two outer sides) will be shifted of a distance according to the thickness of the red edge defined in the technical specification for the triangular road signs [30]. Otherwise if two of the detected sides define the inner triangle, and the remaining one is referred to the outer triangle (figure 3.19b), only two vertexes should be shifted along the internal lines (i.e. those placed on the base of the triangle). When the detected lines do not generate

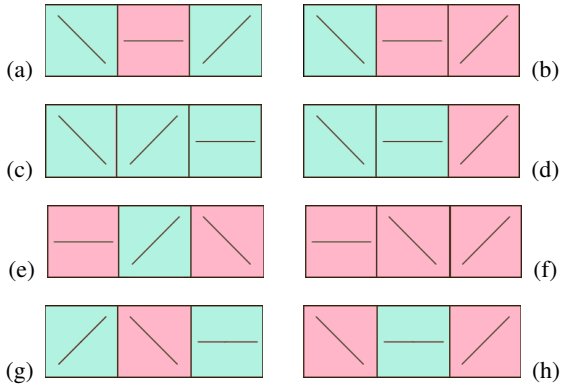
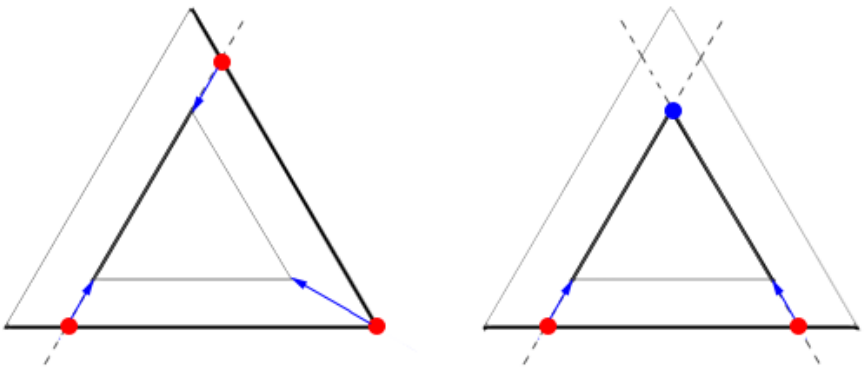
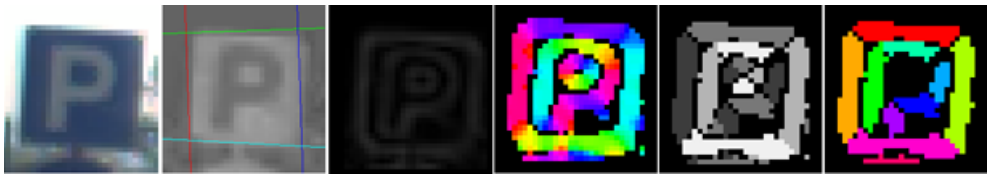


Figure 3.18: Patterns of lines that allow the generation of a triangle. The lines enhanced in red are referred to a side of the inner triangle, the blue ones correspond to an edge of the external triangle.



(a) (b)

Figure 3.19: Detection of: (a) two sides for the outer triangle and one for the inner triangle; (b) two sides of the inner triangle and one of outer triangle.



(a) (b) (c) (d) (e) (f)

Figure 3.20: Processing for rectangular shapes: (a) Rectangular sign to be detected. (b) Lines detection. (c) Absolute Sobel values image. (d) Sobel phases image represented in chromatic scale. (e) Categorization. (f) Labeling and blob detection.

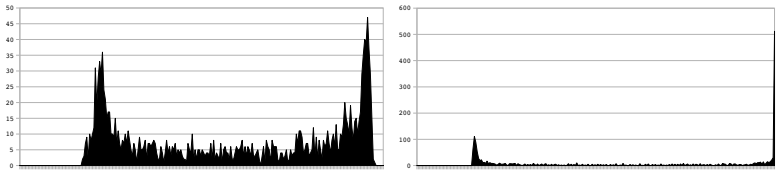
a pattern compatible with the presence of a triangle, bounding box is hold as invalid, considering that it does not frame a triangular shape.

Rectangles

If the analysis of the Sobel phase gap leads to suppose the presence of a rectangle, the same approach used for the triangles detection is applied, unlike the number of sides to search. Since rectangular sign do not have a colored thick border as the triangular ones, it is only necessary search 4 sides, one for each edge of the sign. The categorization (figure 3.20f) is performed considering four possible categories; the obtained lines (figure 3.20b) are valid if their mutual position defines a rectangle.

3.4.4 Considerations on this method

The presented approach allows an improvement of the classification performance, since the crop of the inner area for triangular shapes is obtained considering the edges of the detected sign rather than using a fixed shape mask. This allows to correctly detect a sign also in case that its bounding box is not centered on it. Compared to pattern matching, the algorithm yields better results in presence of rotated and deformed signs, with an increase of only 10% of execution times; anyway the computational overhead requested ensure the real-time application of the system. Compatibly with the acquisition system, in a typical setup the algorithm is able to detect a road sign



(a) (b)

Figure 3.21: Illumination histogram of region provided by detection stage, before (a) and after (b) contrast stretching.

from a maximum distance of 30 meters until the vehicles pass it. Since the performed analysis of the Sobel edges has demonstrated that the phase distribution of the real circular signs does not follow the ideal trend, described method is not suitable for circular sign detection: then futures development will regard the recognition of this type of sign; a possible approach could be that described in [31], that is based on the using of the generalized Hough transform.

3.5 Classification

Regions provided by the sign detector are converted to greyscale and resampled to 50×50 pixels. Since on this regions intensity histograms will be analyzed, and nearest interpolation can generate an imprecise histogram, bilinear interpolation becomes necessary during the resampling process.

In road signs composed by a border and an inner symbol (Danger and Prohibition), only the internal region is resampled to the target size, while the other signs are resampled in their entirety. A mask is then applied on the resampled regions in order to remove the background and the border where present. This mask is generated using shape information (for example a circular mask for Prohibitions and Obligation) and adapted to the border.

From this filtered region a 256 bin greyscale histogram is computed. Since a mask has been applied, the number of pixels analyzed is reduced by up to 50%, but

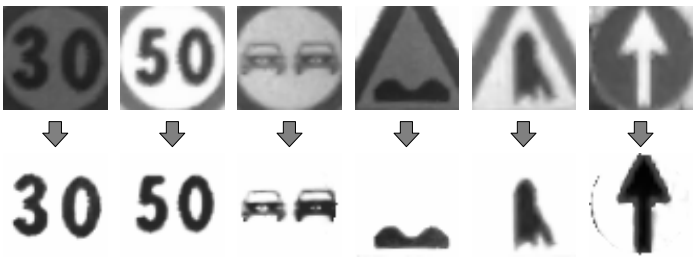


Figure 3.22: Contrast Stretching and filtering applied to some test regions provided by detection stage. Illumination variations are rejected and border of sign partially removed.

it is still large enough to have a reliable histogram without a filtering pass. On this histogram two peaks are selected, one relative to the background and one relative to the shade of the symbol. For road signs with white inner symbol (Obligation), pixel values are inverted, in order to represent all symbols with a black shade and white background, independently of their class. Different sign classes have different ratios between background and inner symbol size, therefore depending on the specific sign class a different peak detector is used to find the shade indexes for the background and the symbol.

The two detected peaks are used to contrast stretch the filtered region between 40 and 255 (see fig. 3.21), with the bottom value (40) chosen according to the black level of synthetic test images. Some examples of refined and illumination corrected regions are shown in fig. 3.22.

The presented algorithm allows to compensate for scale, position and intensity variations, while noise due to occlusions, dirt or other environmental conditions are not handled by the presented processing steps.

The final reduced and normalized regions of any traffic sign candidate are directly used as the input vector of a neural network classifier, which is presented in the following section.

3.5.1 Neural Network

A multi-layer perceptron network with feed forward topology is used for classification [32], and six different neural networks have been trained, one for each reported class of road signs. No special network output is reserved for false positives, since they are expected to provide low values on all output nodes. Networks are trained with the back-propagation method, and validation sets are used to avoid overfitting.

Neural network topologies differ among classes. The input layer is always 50×50 large, and output layer size is equal to the number of signs associated with the given category. The number of hidden layers (one or two) and their size have been chosen to optimize the network performance after extensive benchmarks: the use of only a few neurons and a hidden layer usually does not give satisfactory results, while too many of them causes overfitting. Neural networks with different topologies (with different numbers of layers and layer sizes) have been trained and tested. The best network for each class has been adopted for use in the final application. Currently adopted topologies are shown in table 3.1.

Net Name	*# Hidden Layer*	*Geometry*	*# Output*
Prohibitions	2	115+65	37
Information	1	50	10
Obligation	2	90+40	26
Danger	1	175	42
No-Parking	1	40	3
Stop	1	80	3

Table 3.1: Neural Network Geometry

A back propagation approach is used to train the networks, using both synthetic patterns (roto-translated synthetic signs) and real ones, manually chosen in order to cover a broad range of cases, but trying to avoid excessive specialization of the network: in order to avoid that kind of specialization, to train the networks only a portion (about 5-10%) of real signs is used,

The neural networks are functions with 2.500 (50×50) inputs and several out-

Set Name	*# Pattern*
Synthetic	1253
Good Quality	22119
Poor Quality	2467
False Positive	4381

Table 3.2: Training/Validation Set

puts; each output represents the confidence of classification of the associated sign. Outputs of the network are further processed during the tracking stage described in the following section.

3.5.2 Tracking

the use of a tracking stage is important for several reasons:

- neural networks output is not stable, and can occasionally provide wrong results;
- in order to provide a useful road sign classification system, only one output for each encountered sign should be reported.

Before classification all the regions selected by the detection stage are compared with those classified in the previous frames. For this reason only successfully classified regions can be tracked.

In case of a successful match regions are tracked, while signs remaining without a match are marked as 'ghost'.

Neural networks can be considered as transfer functions with several inputs, and n outputs $O_0 \dots O_n$. In any frame, outputs of the neural network are averaged with the previous network output $A_0 \dots A_n$ of the same tracked sign, using as weight the region size w, so that far candidates have less weight compared to the closer ones.

$$\begin{aligned} A'_i &= A_i + O_i * w \\ W' &= W + w \end{aligned} \tag{3.3}$$

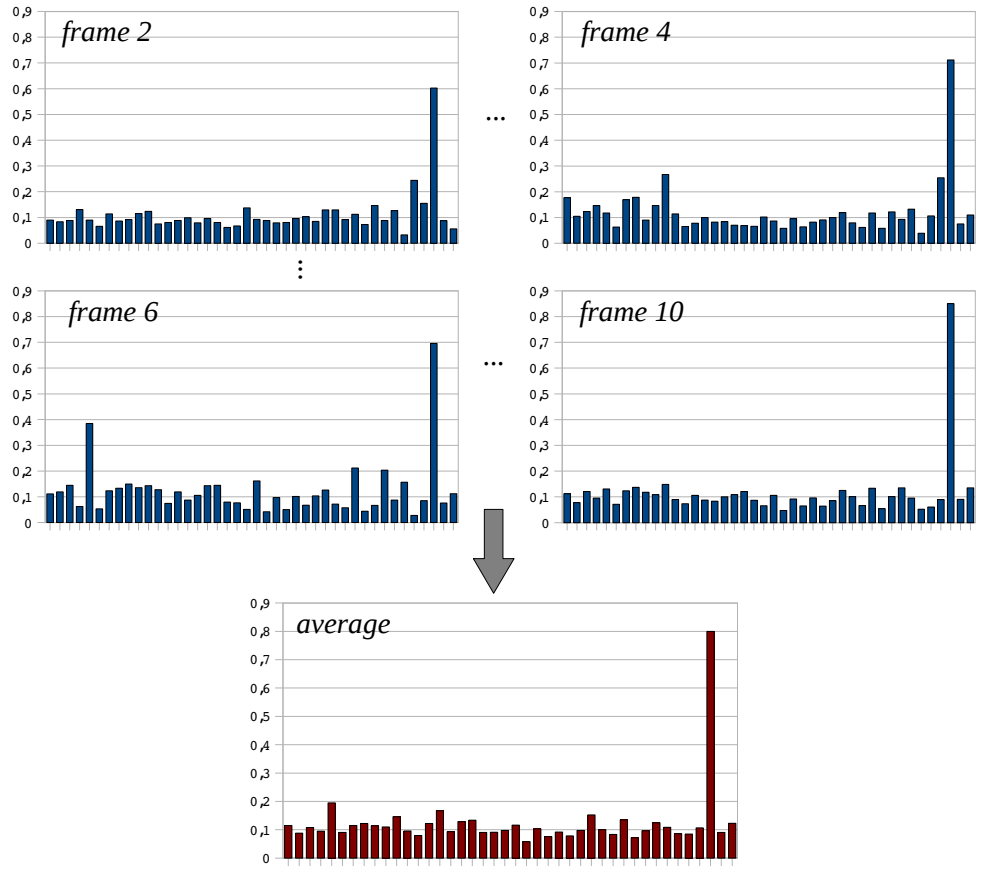


Figure 3.23: Neural Network output averaging. Some outputs of neural network of the same sign tracked among 10 frames are showed, and average of all histograms are reported.

Figure 3.24: Example of training and validation images: Synthetic signs (a), Good Quality signs (b), Poor Quality signs (c).

When a tracked object goes outside of the screen or becomes 'ghost' for more then 5 frames, the sign identifier is reported using the maximum average value

$$id = \max_i A_i \tag{3.4}$$

only if the normalized output value A_{id}/W is above a given threshold ξ, which in general assumes different values for each signs class. Thresholds have been chosen through the use of receiver operating characteristic (ROC) curves analysis.

In order to reduce the system latency, if a road sign is constantly tracked for at least 10 frames, it is classified in advance and directly reported to the user.

Typical response of a neural network processing the same sign among different frames is shown in fig. 3.23. The last histogram in the figure is the averaging of all the responses, filtering out most of the outputs variability.

3.5.3 Experiments and benchmarks

In this section several results produced during the training stage are reported. Results are presented in the form of ROC curves, where correct detection ratio is a function of the false positives number. ROC curves graphically describe the performance of various classifiers, allowing to compare them without the need of explicitly define a

value for the threshold ξ. During network evaluation false positives ratios are computed using both regions wrongly classified to be a sign, and misclassified signs.

All benchmarks presented in this section are produced using the 'Prohibition Set', but the following considerations can be directly extended to other networks.

Sets involved in training and validation of networks can be divided in

- **Synthetic**: signs generated by rotation and translation of a model;
- **Good Quality**: real signs, correctly detected, cropped and normalized;
- **Poor Quality**: real signs, partially occluded, incorrectly cropped, and in general not human readable;
- **False Positives**: false positives reported by the detection stage.

Real patterns used to populate the sets are regions dumped by the system before the classification stage. Examples of signs used during training and validation are shown in fig. 3.24.

The Poor Quality Set is composed by partially occluded signs, saturated signs and any sign hard to classify for a human supervisor. Statistics on this set are made without considering false negatives, but only misclassifications as an error. On this set losing a sign is not important, but it is an error mistaking one sign for an other.

Depending on the sign class, the number and quality of false positives changes. False Positives Set also contains real road signs encountered during the system evaluation but not relevant for the user.

Neural Network Evaluation

Before any further test, the overfitting problem needs to be discussed, since neural networks trained on a limited training set are likely to overfit. To study this behavior a network has been trained for an high number of iterations. After 50.000, 100.000, and 200.000 iterations the network has been evaluated on a validation set, and results are shown in fig. 3.25. When the network starts to overfit the data, the error on the validation set typically begins to rise.

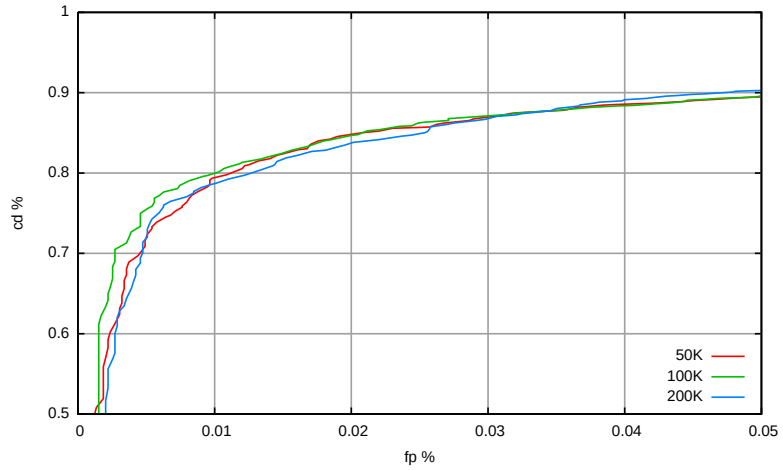


Figure 3.25: Overfit on training network: prohibition network with two hidden layers 115+65 validated after 50K (RED), 100K (GREEN), 200K (BLUE) iterations.

In order to deal with this problem, the training can be stopped when the validation error steadily increases for a specified number of iterations.

To compare the classification results networks are trained using the same training set, composed by synthetic images and a small portion of real traffic signs. Fig. 3.26 shows the comparison between one single layer and two different dual layer networks. The network with a first layer of 115 neurons and a second layer of 65 neurons is the best, and it is actually used in the final application.

Initially training has been made using synthetic signs only; however it did not reach the requested performance. Tests with networks trained on synthetic data are reported in fig. 3.27, with one network trained with synthetic data, one network with real signs and the last with synthetic data and real signs. The training set is composed by 142 synthetic images and 264 real images.

In the works discussed previously, input patterns sizes range from 16×16 to 31×31, while patterns collected for this work are larger. The size chosen is 50×50 pixels in order to easily downsample it for evaluation purpose. In fig. 3.28 comparison

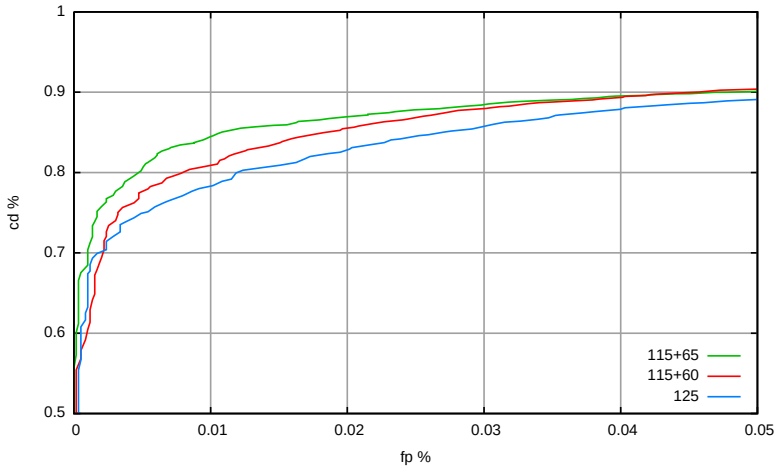


Figure 3.26: Prohibition network trained with network of different hidden layers number and size.

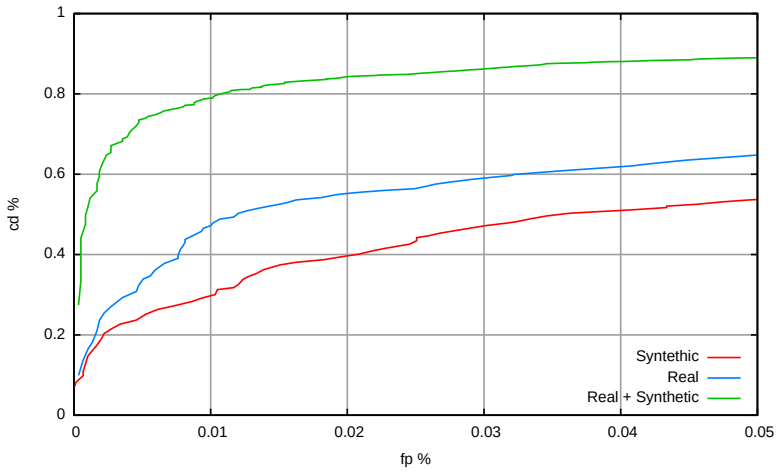


Figure 3.27: Prohibition network with two hidden layers 115+65 trained using only synthetic images (RED), only real ones (BLUE) and a mixture of both (GREEN).

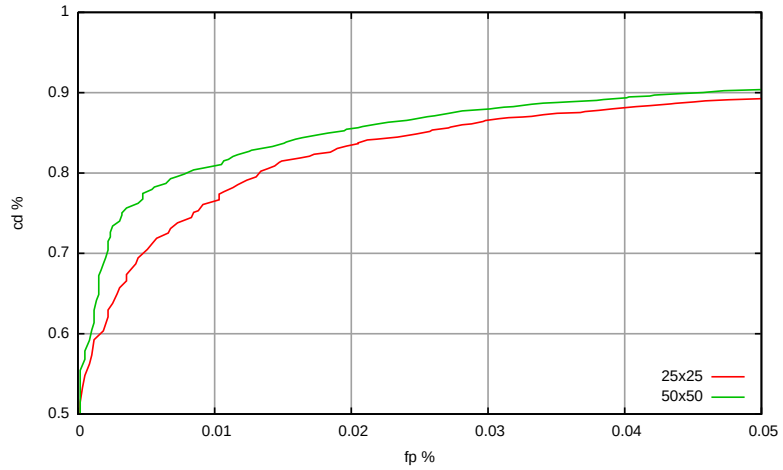


Figure 3.28: Prohibition network with two hidden layers 115+60 trained with input size 25×25 pixels (RED) and 50×50 pixels (GREEN).

between the original network trained on 50×50 pixels samples and one relative to 25×25 downsampled inputs is shown. The 50×50 network is slightly better than the 25×25, however the last one uses 4 times more memory and in a embedded application this occupation can become relevant.

The construction of training and validation sets used in previous tests has been made by a human, which analyzed the worst classified patterns on the validation set and iteratively moved those patterns from the validation set to the training set. Unfortunately, this procedure has proven to be quite slow, inconsistent and can even decrease the performance of the network, as shown in fig. 3.29: this test compares a network trained with 623 patterns selected by a human and a network with 696 randomly selected patterns. Approximately both training set represent 10% of the validation set.

Training set size is still an open issue. Fig. 3.30 shows the performance of networks trained using 10% and 25% of available signs, however the number of signs in the training set is maintained low on purpose to avoid excessive specialization.

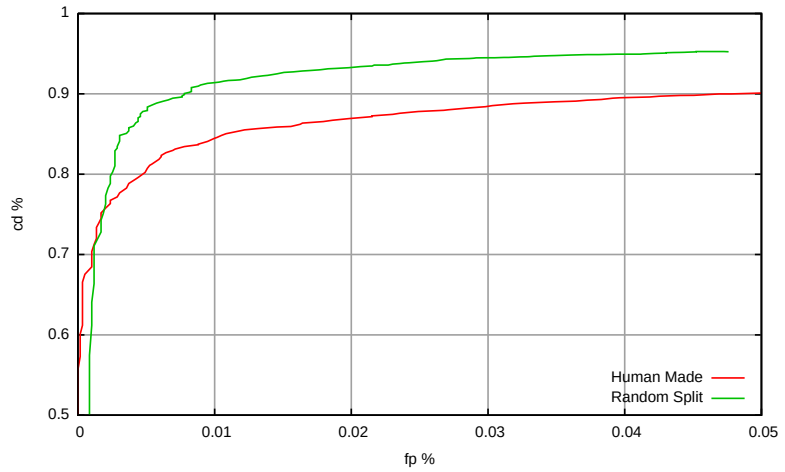


Figure 3.29: Prohibition network 115+65 trained using a train set provided by a supervisor (RED) and random generated (GREEN). Both Set approximately contain the same number of patterns.

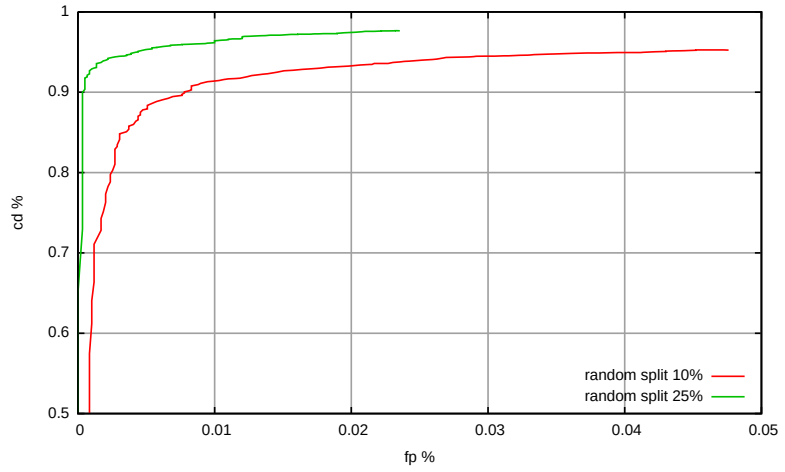


Figure 3.30: Prohibition network 115+65 trained using a train random generated using 10% (RED) and 25% (GREEN) of available patterns.

In fact, the high amount of collected data appears still limited if compared with the high variability of signs appearance. To improve the performance is thus preferable to increase the number of available signs, instead of increasing only the training set size.

Brief benchmarks of networks with thresholds chosen for the final application are presented in table 3.3.

Net Name	*Good Sign CD*	*Overall CD*	*Overall FP / Mismatches*
Prohibitions	83.3%	79.9%	5.9%
Information	87.9%	97.3%	0.5%
Obligation	94.1%	92.2%	3.2%
Danger	76.4%	76.6%	3.2%
No-Parking	95.1%	94.1%	2.7%
Stop	95.7%	94.5%	1.1%

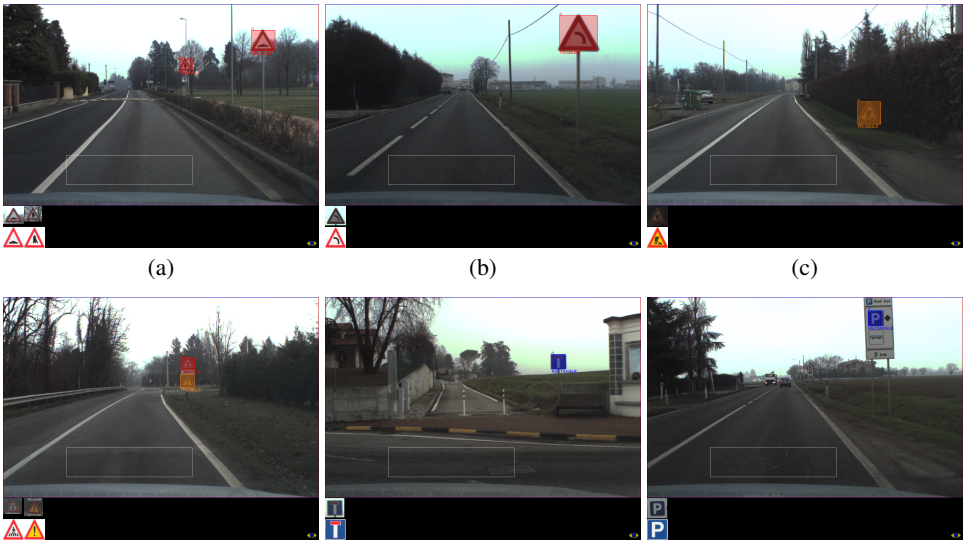
Table 3.3: Brief benchmark of the classification stage.

3.6 Output and results

Tests have been performed in several situations, with different illumination conditions. Figure 3.31 shows some examples of different types of signs. The black stripe placed below each frame is divided in two lines: the top line shows all the signs detected in that frame scaled to a 50×50 pixels image, while the bottom line shows all the corresponding models for the signs that have been classified.

All kinds of sign are correctly detected, even in some ambiguous cases such as figure 3.31.d. Generally, signs are recognized when they are relatively close to the vehicle (e.g. 20 m) and appear not too misaligned with the camera; for example the perspective deformation of the yield sign in figure 3.31.i does not allow the system to detect it, while the same sign in figure 3.31.g is correctly detected. Empirical tests demonstrated that signs can be detected up to 30 meters ahead. However this distance

Figure 3.31: Results on several conditions and different signs. Different bounding box colors indicate primary and secondary color assigned with color segmentation. (a) and (b) pool of two signs; (b) prohibition sign with two colors; (c) red sign with red building on the background; (d) sign on a placard; (e) indication sign; (f) yield sign with a small deformation; (g) yield sign in a saturated image and slightly rotated; (h) obligation sign; (i) yield sign is not detected because of its high rotation, the blue sign instead, has been recognized even if it is mounted upside-down.



(a) (b) (c)

(d) (e) (f)

Figure 3.32: Detection of: (a-b) danger signs, (c-d) work in progress signs, (e-f) rectangular information signs.

(a) (b) (c)

(d) (e) (f)

Figure 3.33: Detection of rotated signs.

(a) (b)

Figure 3.34: False positives.

can be increased reducing camera focal length but dropping the possibility to detect signs that are close and at the side of the vehicle.

During all the development process, low computational time has been one of the issues to follow: on a Pentium 4 at 3 GHz the algorithm runs faster than 10 Hz.

Another open problem is to understand whether a detected sign refers to the driver or not: it can happen, especially in junctions, that a sign is seen by drivers running on another road. This problem can be solved only if the system can perceive the road and junction structure. It is not unusual that a sign is mounted in a wrong way as, for example, shown in figure 3.31.g.

Regarding the shape detection based on Sobel phase analysis, the algorithm has been deeply tested on different scenarios frames leading robust results: all categories of triangular (figures 3.32.a, 3.32.b, 3.32.c, 3.32.d) and rectangular (figures 3.32.e, 3.32.f) signs are correctly detected. The presence of slightly illuminated sign (figure 3.32.c) does not affect the detection performances.

As shown in figure 3.33 the presented approach allows the detection of rotated signs, both in presence of small rotations (figures 3.33.a, 3.33.b, 3.33.c, 3.33.d) and heavy rotations (figures 3.33.e,3.33.f). The system has been tested in both case of vertical rotation, due to a misplacement of the road sign and in case of horizontal rotation that will result in a rotation due to the perspective effect.

In figure 3.34 are shown two cases of false positives where the stripes placed on the trash bin (figure 3.34a) and two trees in front of a red building (figure 3.34b) generate patterns compatible with the presence of a priority sign. The presence of these false positives could be mitigate analyzing the positions of the elements with respect to the road in order to verify that they are set in a place compatible with that of a road sign.

Chapter 4

An example of autonomous vehicle: TerraMax

As an example of the highest level of automation, *Fully automated* driving, in this chapter an example of an autonomous vehicle is presented. In particular the vehicle described, took part to the DARPA Urban Challenge, held in California on November 2007. Being an autonomous vehicle, TerraMax features a complex sensing suite and a decision layer to perform navigation and maneuvering.

First an overview of the whole system is given and of the overall architecture of the robot, then follows a detailed description of one of the perception system developed.

4.1 The DARPA Challenge

The Challenges organized by DARPA are considered some of the most important milestones in the field of vehicular robotics. Not only did DARPA provide a vigorous help in the flourishing of new technological solutions, but the many researchers who accepted these challenges demonstrated that the current technology is indeed mature enough to drive vehicles in off-road and urban-like environments.

The design and realization of autonomous vehicles capable of handling unknown

Figure 4.1: The TerraMax vehicle.

and dynamic situations involve different aspects, spanning from new technologies to implementation, from concept to hardware issues, challenging many different research fields.

A completely autonomous vehicle needs to perceive the environment, assess the situation and react to it, and finally control the actuators to implement a correct behavior. In other words, the vehicle must have perception, decision, and control capabilities. The DARPA Grand Challenge –in particular the second one in 2005 when 5 vehicles were able to reach the end of the 130 miles course,– the DARPA Urban Challenge –which witnessed the completion of the course by 6 vehicles,– and other public demonstrations –such as the PReVENT demonstration at the end of 2007 [33]– showed that the control side has already reached an advanced development stage. Some advanced control systems (such as ABS, EBD, ESP,...) have in fact already reached the market.

Conversely, developing effective and reliable perception and decision systems is definitely more challenging. The perception problem is very complex and even not well defined: a perception system must deal with a high number of different situations and unknown scenarios. Objects in a scene have to be perceived and classified; a cluttered background –typical of a urban environment and therefore of a common situation– can lead to misdetections and introduce a high level of noise in the results.

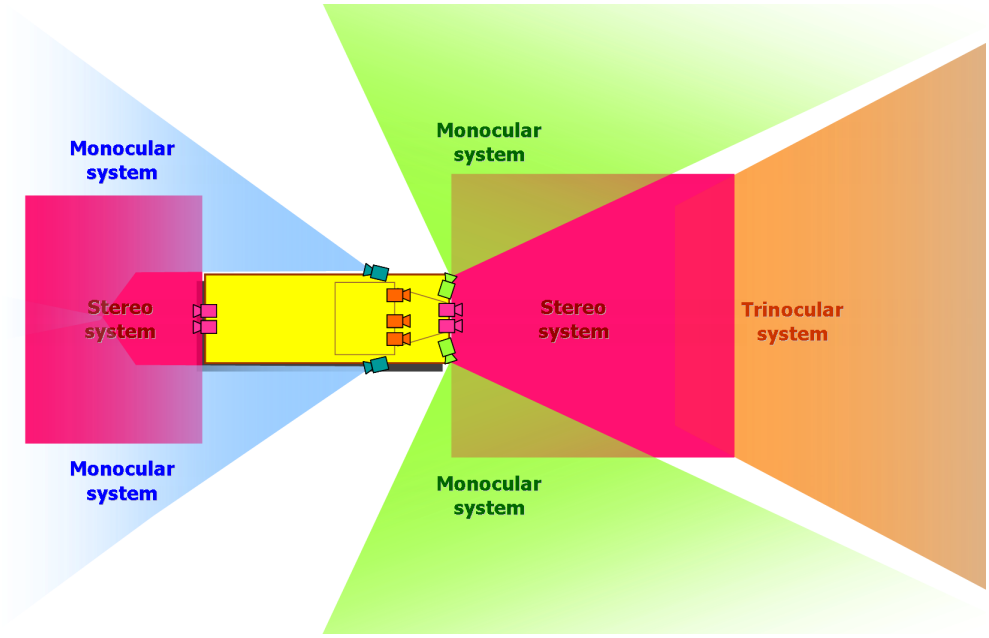


Figure 4.2: The Terramax vision systems (TerraMax facing right) and their placement: orange – trinocular system; purple – stereo systems; green – lateral system; cyan – rear view system.

Similar considerations also apply to decision systems: in order to decide on the action to be taken, it is not sufficient to model the vehicle itself like it happens on the control side, but it is mandatory to model the real world as well; therefore, a huge number of situations must be taken into account.

Different technologies have been tested and implemented on the robotic vehicles that participated in the DARPA Urban Challenge, ranging from conservative products up to completely innovative solutions. Based on the teams' capabilities, budget, and –most importantly– vision, the various vehicles have tackled the perception problem in different ways.

Team Oshkosh's vehicle, TerraMax, was designed to operate with conservative technology so that the integration of the final sensing suite on new vehicles and/or its retrofit to existing vehicles could be straightforward. Furthermore the choice of

using passive technology (sensors not emitting any signal) is generally preferred in the military environment and is also dictated by the will of producing vehicles able to operate in close proximity, namely in situations in which active sensors may interfere with each other.

The final solution was then to adopt cameras as the main sensing device and develop a 360° sensing suite based on artificial vision. Eleven cameras have been mounted on TerraMax to perceive all possible events around the vehicle that may need to be considered by the decision system: presence of obstacles, moving vehicles, lane markings, horizontal road signs. To improve system robustness, three laserscanners have also been installed –two in the front and one on the back– to complement vision perception. Besides providing their own information to the decision system, these 3 LIDARs supply raw data to the vision systems, which perform data fusion to fine tune their final results.

The overall system software architecture, depicted in figure 4.3, consists of two main categories of service modules, System Services and Autonomous Services. System Services include health monitoring, service control and configuration, and visualization. Autonomous services include sensor services, autonomous behavior services, perception services, and vehicle management services. The functionalities of these modules are described in the following subsections.

4.1.1 Autonomous Services

Autonomous Vehicle Manager

The Autonomous Vehicle Manager (AVM) manages the high level operation of the vehicle. It is primarily responsible for performing route planning, trajectory planning, and behavior management. The AVM receives perception updates from the World Perception Server (WPS) and uses this information to track the current vehicle state and determine the current behavior mode. The AVM continuously monitors perceived obstacle and lane boundary information and issues revised trajectory plans to the Autonomous Vehicle Driver (AVD) through Drive Commands.

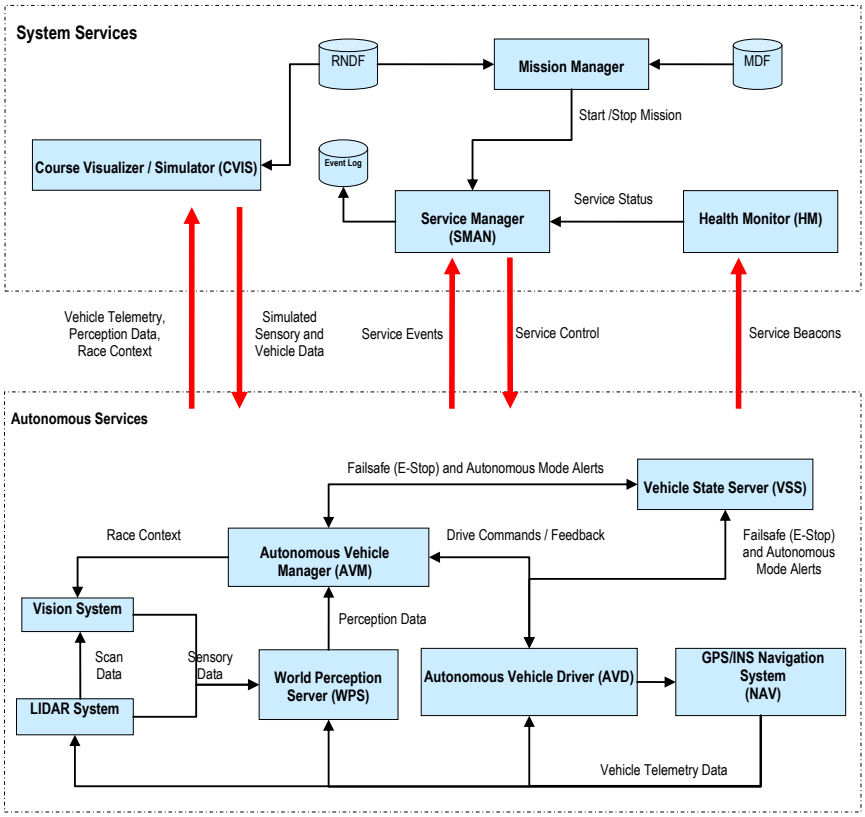


Figure 4.3: TerraMax overall system software architecture.

Autonomous Vehicle Driver

The Autonomous Vehicle Driver (AVD) performs waypoint following, lateral, longitudinal and stability control by accepting messages from the AVM and commanding the lower level x-by-wire controls.

Vehicle State Server

The Vehicle State Server (VSS) monitors, manages and reports all system mode preconditions in the low level control that allow a transition from manual to autonomous operation. The VSS detects any faults, attempts to recover, and transitions the system into failsafe mode if necessary.

World Perception Server

The World Perception Server (WPS) publishes perception updates containing the most recently observed vehicle telemetry, obstacle, and lane/road boundary information. The WPS subscribes to sensory data from the LIDAR and VISION systems. The WPS combines the sensory data with the vehicle telemetry data received from the navigation service (NAV). Obstacles detected by the LIDAR and VISION systems are further fused in order to provide a more accurate depiction of the sensed surroundings. The AVM consumes the perception updates published by the WPS and uses this information to determine the next course of action for the vehicle.

GPS/INS Navigation System

The GPS/INS Navigation System (NAV) Service manages communications to the GPS/INS unit and converts latitude/longitude outputs to UTM coordinates.

Vision System

The Vision System (VISION) processes vision data obtained from 11 cameras in order to sense the environment around the vehicle. A detailed description of this module is given in the following sections.

LIDAR

We have integrated a COTS LIDAR system that provides a 360° field of view. The LIDAR system software has been adapted to operate within our system architecture messaging schema. Each laserscanner scans a 220° horizontal field. Two laserscanners are positioned on the front corners of the vehicle and one laserscanner is positioned in the center of the rear. The front scanners are fused at the low level; the rear system remained a separate system. Obstacle detection and tracking is performed in the system ECUs and published to the WPS, while raw scan data is used by the vision systems to enhance precision and aid in identifying interest areas.

4.1.2 System Services

Health Monitor

The Health Monitor (HMON) maintains the status of all services by monitoring service beacons that are sent as UDP multicast data packets. Each service is required to send a service beacon at a prescribed periodicity. The service beacon contains the services status, service port, and IP address. The Health Monitor sends alert notifications when a service has reported an error or when a service has not sent a beacon within the allotted timeframe. The Service Manager will respond to the health monitor alert notifications and take appropriate measures; such as restarting an unresponsive service.

Service Manager

The Service Manager (SMAN) manages the initialization, startup, and shutdown of all autonomous services. During system startup, the Service Manager orchestrates the startup process using service dependency information that is read from a configuration file. The Service Manager starts or stops the autonomous service in response to the mission start/stop notifications sent by the Mission Manager application. The Service Manager maintains the mission status so that autonomous services can be properly configured and restarted in the event of system reset. The Service Manager

monitors health alerts originating from the Health Monitor. When a service is no longer responding or has indicated an error, the Service Manager will take appropriate actions, such as attempting to restart the service or causing a system reset.

Mission Manager

The Mission Manager (MMAN) provides the user interface for configuring and starting the autonomous services in preparation for autonomous mode. The application allows a user to load the Road Network Definition File (RNDF) and Mission Definition Files (MDF) files. The Mission Manager validates the RNDF and MDF files before loading them into the system. To start a mission, the user clicks on the "Start" button. This instructs the Service Manager to initialize and start all autonomous services.

4.2 TerraMax Vision Systems

TerraMax perception is provided by 11 cameras grouped into 4 different systems, each supplying information about the environment in a specific area around the vehicle. Fig. 4.2 shows a sketch of the coverage of the vision systems.

A trinocular vision system (orange in Fig. 4.2) is used to perceive information about the road in front, including lane markings, road borders, and possible obstacles/vehicles on the driving path. Two stereo vision systems (purple in Fig. 4.2) monitor the close proximity of the vehicle's front and back in order to locate obstacles and road signs. The precise localization of road signs (lane markings and stop lines) is essential due to the vehicle's abnormal size. Two monocular vision systems (green in Fig. 4.2) are used to surveil the sides of the vehicle when approaching intersections; normally off, these two high resolution cameras start to process images as soon as the vehicle stops at an intersection, and look for oncoming traffic in any possible direction (not only orthogonal to the vehicle direction). Finally, two other monocular vision systems (cyan in Fig. 4.2) keep track of overtaking traffic in adjacent lanes; their output is required when TerraMax needs to change lane.

Table 4.1: TerraMax subsystems summary

	TRINOCULAR	**STEREO**
Cameras	3 cameras 1024×768	4 cameras 1024×768
Cameras Position	Upper part of the windshield, inside the cab	Two on the front camera-bar, two on the back of the truck, all looking downwards
Optics	focal length 4.16 mm, f2.16, FoV: 60°×42°	focal length 1.8 mm, f1.4, FoV: 154°×115°
LIDAR	Front	Front, back
Algorithms	Lane detection, stereo obstacle detection	Lane detection, stop line detection, curb detection, short-range stereo obstacle detection
Range	7 to 40 m	0 to 10 m and -6.5 to -16.5 m
Notes	3 stereo systems with baselines: 1.156 m, 0.572 m, 1.728 m	2 stereo systems (front and rear)

	LATERAL	**REARVIEW**
Cameras	2 cameras 1920×1080	2 cameras 1024×768
Cameras Position	On the sides of the front camera-bar	External, on top of the cab, looking backwards and downwards, rotated by 90°
Optics	focal length 8 mm, f1.4, FoV: 80°×63°, for HiRes sensors	focal length 4.16 mm, f2.16, FoV: 60°×42°
LIDAR	Not used	Back
Algorithms	Moving traffic detection	Overtaking vehicles detection
Range	10 to 130 m	-4 to -50 m
Notes	Enabled when the truck stops at crossings	Monitors the adjacent lanes to allow/inhibit lane change maneuvers

Figure 4.4: Hardware arrangement inside the passenger seat for the various systems.

All the vision systems share a common hardware platform consisting of a rugged PC from SmallPC[1], equipped with a mini-ITX board from Mini-Box.com[2], an Intel[3] Core 2 Duo T2500 CPU, 2 GB of RAM and an Intel 945GM chipset. Two FireWire 800 boards, one PCI, and one PCI Express provide 4 connectors on the back. All the PCs are placed under the passenger seat inside the cabin, together with LIDAR, INS/GPS, and vehicle control units, as shown in Fig. 4.4.

The imaging hardware for the trinocular, stereo and rear systems is based on Point Grey Research[4] Flea2 industrial cameras, equipped with a 1/3", 1024×768 pixels Sony CCD sensor, while the lateral one features two full HD (1920×1080), FireWire B, AVT[5] Pike2 industrial cameras; the exposure parameters, like gain and shut-

[1] http://www.smallpc.com
[2] http://www.mini-box.com
[3] http://www.intel.com
[4] http://www.ptgrey.com/
[5] http://www.alliedvisiontec.com


(a)

(b)

(c)

Figure 4.5: Stereobox mounted on (a) Volvo and (b) Iveco trucks: a red circle highlights cameras position. (c) shows VOD, the embedded version of StereoBox.

ter speed, can be adjusted through the FireWire interface, while synchronization at 12.5 Hz with the three IBEO[6] AlascaXT LIDARs is provided by a dedicated SyncBox. Cameras are connected to the processing system using 7.5 m cables from AVT, specifically designed for industrial use in noisy environments. Cables, cameras and boards are also equipped with locking screws to keep the connectors in a fixed position, even in critical environments with strong vibrations; all cameras placed outdoor are enclosed in sealed boxes, in order to preserve them from dust and rain. Polarizers have been used to minimize the reflections.

Table 4.1 summarizes the hardware setup.

4.3 The stereo system

Short range perception is one of the main issues in both autonomous driving and driving assistance systems because most of the dangerous situations for the driver himself and for other vulnerable road users take place at low speeds in urban scenarios. In particular, short range perception provides to autonomous vehicles a precise knowledge of the surroundings, allowing them to plan safely their movements in the environment; when used for driving assistance, it can assure correct obstacle de-

[6] http://www.ibeo-as.com

Figure 4.6: The front stereo system mounted on the TerraMax vehicle.

tection, a fundamental aspect for collision mitigation and parking assistance systems. The system used on TerraMax is derived from the StereoBox [34] a short range obstacle detection system. StereoBox has been successfully fielded on several prototypes, such as Volvo and Iveco trucks, as a Start Inhibit system (figure 4.5). The two color cameras monitor the blind area in front of the truck stopping vehicle motion in case of obstacles. StereoBox's technology and algorithmic solution have been successfully validated thanks to an accurate and exhaustive testing period after which StereoBox was ported on a DSP based embedded system, called VOD (VisLab Obstacle Detector), to meet carmakers and automotive system providers requirements of a realtime, compact and low cost system (figure 4.5.c). Two twin systems have been developed for the DARPA Urban Challenge, StereoFront and StereoBack, which further extend StereoBox applications. Together with the obstacle detection system, provided by the StereoBox, StereoFront and StereoBack perform lane markings detection and data fusion with information provided by additional laserscanners installed on TerraMax.

After having seen five vehicles reaching the finish line in 2005, the Defense Advanced Research Project Agency (DARPA) moved its third robot race from the desert into a city environment, calling it Urban Challenge. The Urban Challenge features autonomous ground vehicles maneuvering in urban and sub-urban scenarios, where

they had to execute merging into moving traffic, navigate traffic circles, negotiate busy intersections, avoid obstacles, follow lanes, and handle parking. VisLab provided the vision system to sense the surroundings of the vehicle; it was composed of 11 cameras to get full 360 degrees (all-round) vision and detect lane markings, stop lines, the drivable path, vehicles, and obstacles, for automatic forward and backward driving, lane changing, maneuvering and managing intersection.

Since the trinocular system can only be used for perception at medium-to-long distances; to extend its detection capabilities TerraMax includes two stereo systems (one in the front and one in the back), which provide a precise sensing of obstacles and lane markings in the close proximity of the vehicle. The two stereo systems are identical from both the hardware and software standpoints: the 4 cameras are connected to the same PC which selects the system to use based on driving direction and speed (forward, backward, or both when the speed is close to 0 km/h). Thanks to wide angle (fisheye, about 160°) lenses from Fujinon[7], the sensors gather information on an extended area of the immediate surroundings, about 10×10 m.

The software is designed to improve the detection of other vision systems: the stereo systems are designed to detect obstacles and lane markings with a high confidence on the detection probability and position accuracy.

Calibration for the stereo system has to cope with highly distorting cameras are used without any knowledge about the intrinsic and extrinsic camera parameters. An analytic approach to calibration would be computationally prohibitive: the equations that are normally used to model spheric lenses become too complex when wide-angle lenses are used. Thus, an empiric approach has been used, and it is described in the following section.

4.4 Calibration

Camera calibration is one of the most important topics for vision systems especially when fielding systems that must be installed on real vehicles which have to operate in real scenarios.

[7] http://www.fujinon.com/

During an offline preprocessing a look-up table that allows a fast pixel remapping is generated; namely each pixel of the distorted image is associated to its corresponding pixel on the undistorted image. Images of a grid, painted on a stretch of flat road in front of the truck, are used to compute the look-up table (see figure 4.7). A manual system to pin point all the crossing points on the source image is used.

Thanks to the knowledge of the relative position of the truck with respect to the grid itself and to the assumption that the road can be considered nearly flat in the proximity of the vehicle, it is possible to compute a new image (the *IPM* image) removing both the perspective effect and camera distortion at once. A non-linear interpolation function is used to remap the pixel of the source image that are not cross points.

The process to determine coordinates (x,y) of the source image from the (i,j) pixels of the IPM image is divided in two steps.

Let's assume to have a grid with N vertical lines and M horizontal lines. For each vertical line of the grid, a function f_n is defined, where $n \in [1,N]$ is the line number. The spline creation is constrained by the correspondences between the crossing points of each line in the source image and in the IPM image; see equation (4.2) as an example, assuming x_1, y_1, x_2, y_2, etc. as the coordinates of the crosspoints on the source image.

$$f_n(j) : \Re \longrightarrow \Re^2 \qquad f_n(j) = \begin{cases} f_n^x(j) \rightarrow x \\ f_n^y(j) \rightarrow y \end{cases} \tag{4.1}$$

$$\begin{cases} f_1^x(0) = x_1 \\ f_1^y(0) = y_1 \\ f_1^x(1) = x_2 \\ f_1^y(1) = y_2 \\ \vdots \\ f_1^x(N) = x_N \\ f_1^y(N) = y_N \end{cases} \tag{4.2}$$

Using functions $f_1(j)$, $f_2(j)$, ..., $f_N(j)$ another class of functions can be built,

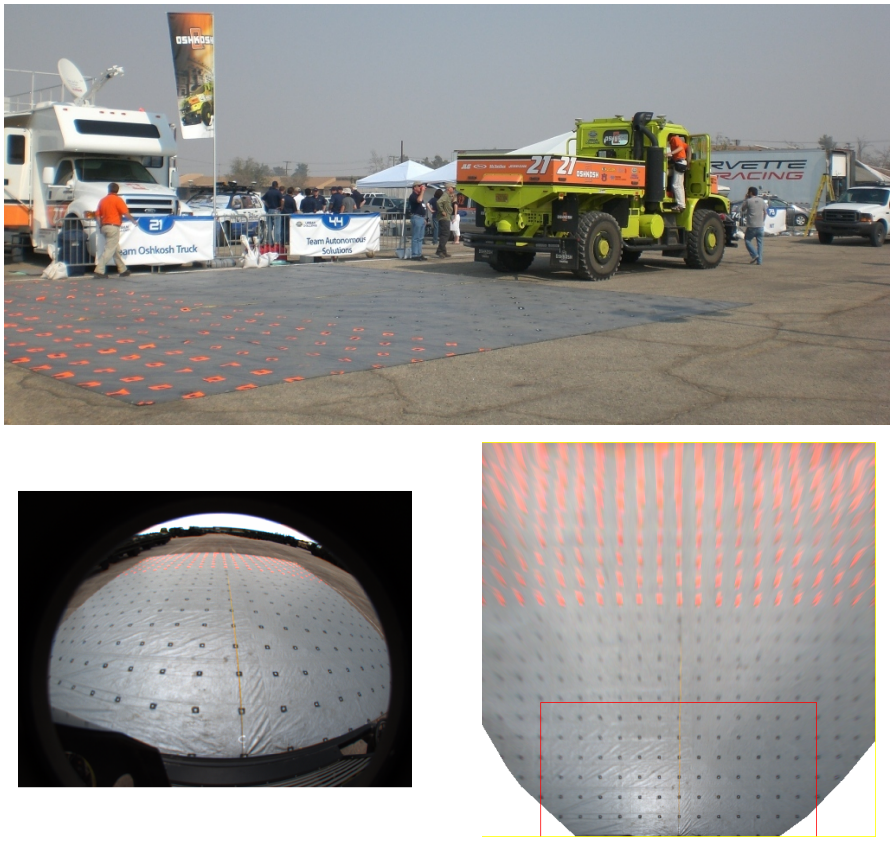


Figure 4.7: Stereo system calibration is performed by means of a grid painted on a portable tarp. The picture on the left shows TerraMax while calibrating the stereo back, then is shown the acquired image and the result after the pixel remapping using a lookup table.

called $g_j(i)$ and defined as described in equation (4.3) with equation (4.4) as constraint.

$$g_j(i) : \Re \longrightarrow \Re^2 \qquad g_j(i) = \begin{cases} g_j^x(i) \rightarrow x \\ g_j^y(i) \rightarrow y \end{cases} \tag{4.3}$$

$$\begin{cases} g_j(1) = f_1(j) \\ g_j(2) = f_2(j) \\ \vdots \\ g_j(N) = f_N(j) \end{cases} \tag{4.4}$$

In this way all the pixels of the IPM image have a correspondence to a pixel of the source image and the cubic spline interpolation method allows to get the best match between the two sets of pixels.

Being the system based on stereo vision, two tables, one for each camera and both fixed under the same reference frame, are computed with this procedure. The look-up table generation is a time-consuming step, but it is computed only once when the cameras are installed or when their position is changed.

4.5 Obstacle detection

Obstacle detection, derived from the Start Inhibit algorithm [34], is performed in two steps: first the two images, acquired simultaneously, are preprocessed in order to remove the very high lens distortion and the perspective effect, and then the two images are matched so that any appreciable deviation from the flat road assumption is labeled as an obstacle. The flat road assumption is considered valid since the area of interest is limited to a maximum of 8 to 10 m. Thanks to this assumption, the system:

- is able to detect any kind of obstacle without a-priori defined obstacle classes,
- is not based on obstacle motion, which usually generates false positives when the vehicle moves,

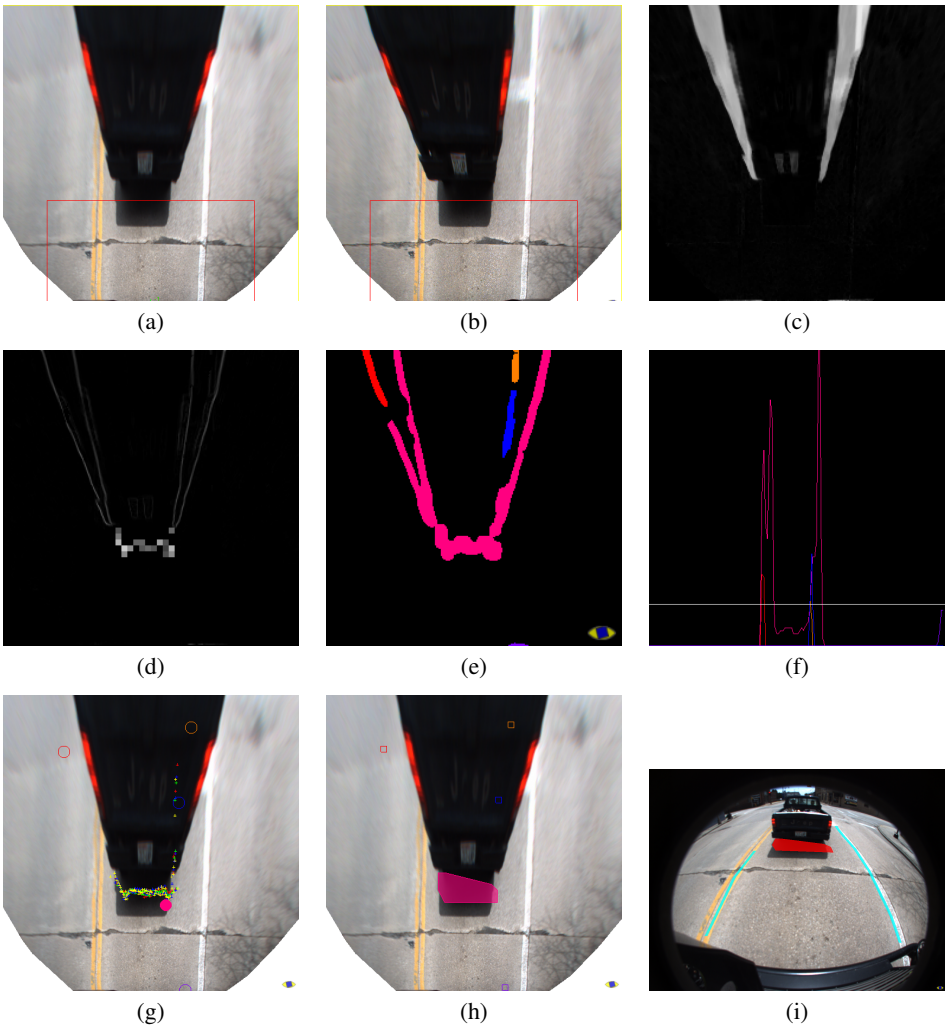


Figure 4.8: Processing steps of obstacle detection. (a) and (b) IPM images; (c) difference image between the two IPMs; (d) laser clusters added to the difference image after sobel filter application; (e) blob extraction; (f) polar histogram; (g) and (h) laser data remapped on IPM before and after clustering; (i) final result remapped on one of the source images.

Figure 4.9: All the Terramax sensors highlighted. Color coding: red – cameras; green – laserscanners; blue – GPS antennas.

- is robust with respect to shadows on the ground since it performs a comparison between images taken at the very same time and since the texture generated by shadows belongs to the road plane.

Therefore a stereo inverse perspective mapping-based approach has been considered [35]. The whole processing is performed by means of two main steps:

- lens distortion and perspective removal from both stereo images,
- obstacle detection.

Concerning the first step, the problems of distortion removal and inverse perspective mapping (IPM hereinafter) without the knowledge of the intrinsic and extrinsic parameters of cameras have to be solved. Lens distortion is usually modeled as polynomial radial distortion and it is removed by estimating the coefficients of this polynomial. After the distortion removal phase, extrinsic parameters are obtained, nevertheless, the highly complex mathematical model of the spherical lens may affect the computational time. Therefore, a graphic interface to remove lens distortion has been designed to manually associate the grid points of the source image to their homologous points on a square grid on the IPM image. This preprocessing is performed offline and the result are stored in a lookup table for a quicker online use.

Starting from the two IPM images (figure 4.8.a and b), the difference is computed and a filter is applied to remove small differences due to possible small miscalibrations (figure 4.8.c). At this step, data coming from laserscanners are clustered [36] and then overlayed to the difference image so that laser reflections in a particular region can boost the difference image and thus trigger the presence of an obstacle, see figure 4.8.d. Connected areas appearing in the resulting image are localized and labeled: a progressive number is assigned to each label for further identification.

A polar histogram is computed for each region (figure 4.8.f). The focus used to compute the polar histogram is the projection of the mid point between the two cameras onto the road plane. These regions produce peaks on the polar histogram. Thus, the presence of strong peaks can be used to detect obstacles. Then a filter to the polar histograms to remove regions that cannot be considered as obstacles is

applied. This filtering is performed considering the width of the histogram for the region of interest. The width of the histogram is computed in correspondence to a given threshold. When a polar histogram features several peaks, different values of width (w_1, w_2, etc.) are generated. If $\max\{w_1, w_2, ..., w_n\} > w_{min}$ (where w_{min} is a width threshold), then the region previously labeled is maintained, otherwise it is discarded.

4.5.1 Laser data fusion

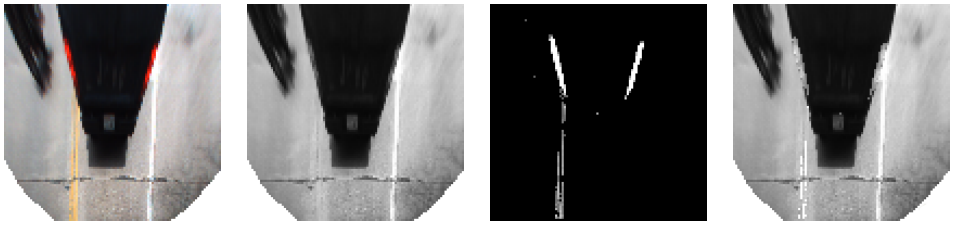
The system developed for the TerraMax vehicle [37] has been improved with a laser-scanner-vision data fusion since 3 LIDAR devices were mounted on board: two in the front and one in the back as shown in figure 4.9.

The sensors output both processed data (i.e. a list of detected obstacles) and raw data: while the former is useful for the high-level path planner, the latter is exploited to improve the performance of the various vision systems.

Vision and LIDAR systems are complementary in many ways: for example, when performing a tridimensional reconstruction of the world LIDAR points offer a very accurate, yet sparse representation of the world, while vision provides dense, but less accurate measurements. Some tasks, like lane and path detection, are also better accomplished through the use of vision. Integrating the data coming from the two sensors in useful, non-trivial ways is a challenging operation, but while it poses additional problems, like multi-system calibration (which must be performed accurately in order to collect consistent data, e.g. obstacle positions), it can also significantly improve the overall system behavior.

The final goal has been to achieve a vision system set that can reliably perceive all round the vehicle even without any fusion with other sensors. Data fusion helps the overall system to manage some particular situations and to increase the information and accuracy of the output produced.

The cameras setup selected for this system (fisheye lenses) has the drawback of making it difficult to locate the exact boundaries of the detected items; to address this shortcoming, laser data is clustered (figure 4.8.h) and matched against the detected elements in the field of view, and when a correspondence is found the precise shape is



(a) (b) (c) (d)

Figure 4.10: Lane detection preprocessing step. (a) color input image; (b) grey level conversion; (c) yellow enhancing filter application; (d) enhanced grey level image.

associated to the obstacle. If no laser data is available, or the matching phase produces poor results, the system provides only the position of the obstacles, with no additional shape information.

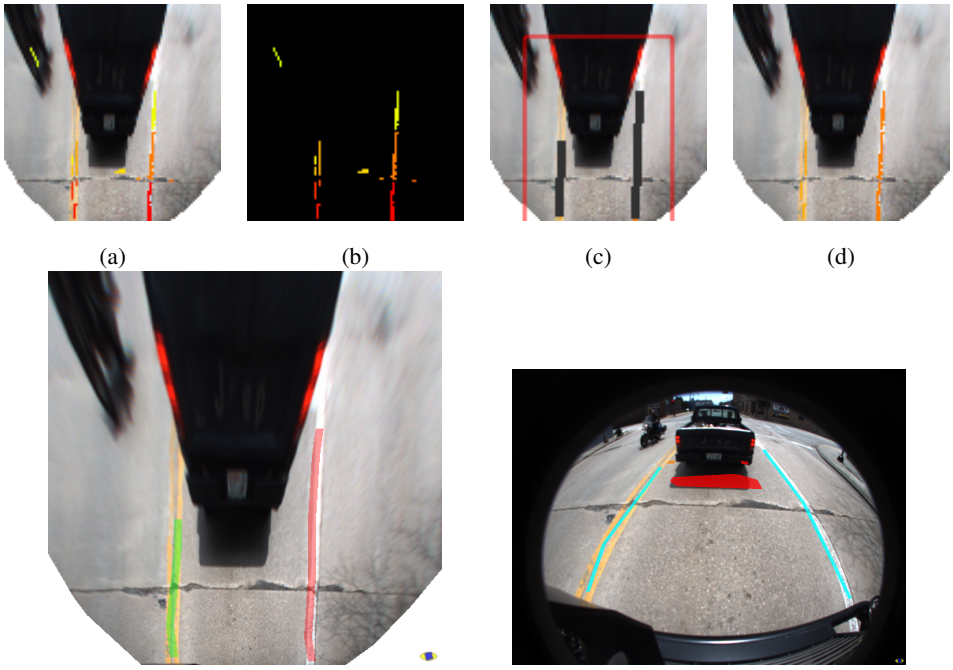
4.6 Lane markings detection

The other main functionality of this system is lane markings detection starting from a monocular image: the detection is focused on lane markings, stop lines, and sharp curves thanks to the wide field of view.

The input image is one of the 2 IPM images coming from the previous obstacle detection processing. Processing is divided in 3 main steps: preprocessing, processing, and tracking, all described in the following.

4.6.1 Preprocessing

First the input image is rescaled to focus only on the region of interest, then a grey-scale conversion is applied. Since lane markings can be yellow and any grey scale conversion deeply reduces the contrast between yellow lines and grey asphalt, a particular color enhancement filter is applied; to save computational power this filter has been coded into a look up table with Red and Green values as indexes. The resulting image, is used as a mask to enhance yellow pixels in the original image converted in grey-scale (see figure 4.10).



(a) (b) (c) (d)

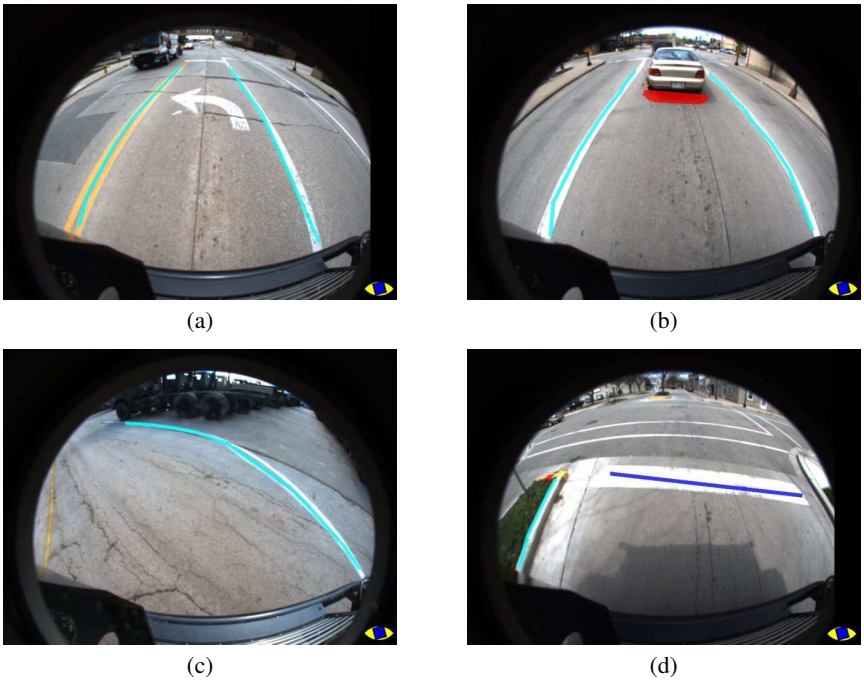
(e) (f)

Figure 4.11: Lane detection processing step. (a) and (b) labels found on the image (each label is encoded with a different color); (c) fusion of adjacent labels; (d) final result; (e) and (f) final result remapped on IPM and on the source image, (f) includes also obstacle output.

4.6.2 Processing

The processing is based on the construction of a LHL image: it means that a filter searching for a Low-High-Low patterns on grey levels is applied. To avoid shadow effects or errors due to different asphalt colors, the pattern is searched considering also the neighborhood of each the pixel; in particular a low pass image is precomputed.

Then a labeling step is applied. Each label is recursively expanded following its orientation, searching for other labels to fuse. After the fusion stage, each label is



(a) (b)

(c) (d)

Figure 4.12: Output examples. (a) lane detection; (b) lane and obstacle detection; (c) lane detection on a sharp curve (90° curve); (d) stop line detection.

evaluated in order to discard lines with fast orientation changing that probably do not belongs to the road surface. Finally the longest label on the left side and the longest on the right side are chosen, see figure 4.11.e and f.

A tracking stage has been also added at the end of the processing to increase stability and reliability: a TTL (time to live) is associated to each final line. If a line has been detected for more than k times it is provided as output, otherwise TTL is decreased. When TTL reaches 0, the line is deleted from the tracker memory. Filters on lane width are applied as well to avoid false positives

Stop line detection processing is very close to line detection: preprocessing on yellow is not necessary since stop lines are supposed to be white. LHL filter is computed in the vertical direction and is tuned on wider lines.

Figure 4.13: Currently tested applications of VOD system in different scenarios.

4.7 Output and results

The stereo obstacle detection system is derived from an already developed system mounted on several prototypes and already embedded on a DSP platform. Fish-eye lenses and the reduced height of the system due to installation constraints challenged the problem, but Stereo proved to be robust enough for the purpose.

Data fusion with laserscanners was added to vision obstacle detector, to enhance the algorithm and give to the vision system a higher precision in the distance estimation.

Some problems were encountered for little miscalibrations. Thus a particular filter to remove noise coming especially from cameras miscalibration has been developed. Similarly another filter has been developed to remove some fake obstacles generated by laserscanner scan data improving data fusion performance. After the Site Visit, an ID was given to obstacles in order to help the obstacles association on subsequent updates. The lane detection subsystem was derived from the one implemented

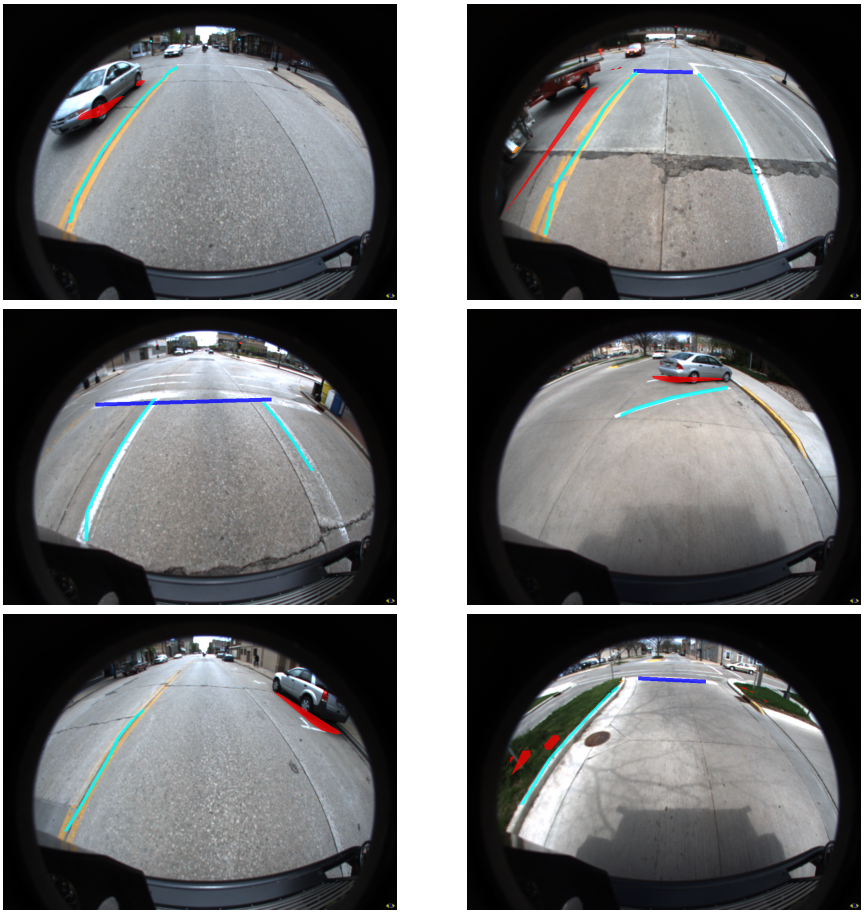


Figure 4.14: Results in a urban scenario. Red boundaries highlight obstacles, light blue lines show lane markings while dark blue lines indicate stop lines.

for the Trinocular system, the other front sensing system placed inside the cabin, as shown in figure 4.9: due to its particular placement, the Trinocular suffered hood masking in the close proximity of the vehicle. Tasks such as stop line detection and lane detection when approaching a sharp curve has been addressed with the stereo system, as shown in figure 4.12, that could handle a better sensing in the close area. Stop lines detection was developed for the site visit and worked very reliably: the placement precision during stop line approaching was evaluated by DARPA officers during the qualification and the race.

The system proved to be reliable and stable in various situation. As shown in figure 4.14 both yellow and white line are detected. Oblique lines are detected as well: this feature has been added to help the vehicle during parking maneuvering. Due to its huge size, TerraMax had to be very precise in driving and the Stereo system could help providing a sufficiently accurate description of the close surroundings.

One of the key feature that allowed this system to be successfully mounted on several prototypes is the very simple calibration, a nice solution is to paint the calibration grid on a tarp to make it easily portable. What has been significantly eased with respect to a disparity based stereo approach [38] is the camera placement: in this case it is enough that the two cameras frame the area of interest without any fine orientation tuning. This facility is paid with a loss in object classification, but for many application (Start Inhibit for example) this lack can be certainly accepted.

One of the most important future improvement is the autocalibration feature: it will be available soon and allows a very quick and easy installation of VOD on commercial vehicles. Even in case of camera misalignment caused by an impact, the system would be able to autodetect a miscalibration and thanks to a simple procedure it could be recalibrated in any authorized center.

Figure 4.13 shows several scenarios in which VOD system can be applied: as a Start Inhibit monitoring the vehicle front or a system monitoring the truck side preventing accidents during turning maneuvers. All the cited areas are blind for the driver and need to be monitored. If mounted on the rear of a car, VOD can be used also as parking assistant system.

Chapter 5

Conclusions

In order to develop the concepts described in this thesis, VisLab set up BRAiVE, (figure 5.1) a prototype vehicle integrating various sensing technologies. A new architecture (UPDA[1]) is meant to take as input all the data coming from the different perception modules, fuse them, and build a more sophisticated world model to be used in the following decision level. In doing this, a standard interface will be defined for each module, allowing different providers to integrate their own perception system into this standard architecture, which will also boost competition.

Providing a UPDA involves the understanding of what has to be perceived and classified by the vehicle. We make a list of the main items in the following:

- pedestrians: still or moving and, if moving, how fast;
- vehicles: and their speed;
- other obstacles: and their speed;
- lanes: any sort of roadways demarcation line: temporary or permanent, continuous or dashed;
- stop lines;

[1] Unified Perception & Decision Architecture

Figure 5.1: BRAiVE: prototype vehicle.

- junctions: crossings layout;
- parking lots: available parking areas delimited by horizontal demarcation lines;
- tunnels and bridges;
- road signs: danger, priority, prohibitory, mandatory and indication signs;
- traffic lights:
- free space: surrounding environment areas available for the vehicle to move and steer;
- blind spots: areas of the road that cannot be seen while looking forward or through either the rear-view or side mirrors;
- light and visibility conditions: daylight, nighttime, dazzle, rain, fog, snow, etc.;
- road slope: road plane inclination;
- environment: can be urban or extra-urban; if the latter, no lanes will be available for the vehicle to find its way.

The advanced driver assistance systems currently available on the market can be described in terms of the items just listed: Lane Departure Warning involves the

detection of lanes and free space; Blind Spot Monitoring the detection of vehicles; the ACC functionality is allowed by detecting the preceding vehicle and Collision Mitigation involves the detection of vehicles, other objects, tunnels and bridges, free space.

BRAiVE will be provided with driver assistance systems similar to those already available on the market but with improved performances, plus other major capabilities, listed in the following:

- Crossings Assistance: lateral perception is important when facing an intersection, and as junctions may have very different layouts, the lateral system must be able to perceive traffic coming from different angular directions. During a traffic merging manoeuvre, the vehicle is allowed to pull into traffic only when oncoming traffic leaves a gap of a sufficient time interval (usually at least 10 seconds). Hence the vehicle, regardless of the junction layout, needs to perceive the oncoming traffic as well as estimate cars speed from long distance.

- Overtaking Assistance: when driving on a road with multiple lanes, the lane change manoeuvre is a common task in order to overtake slower traffic. Two rear cameras installed acquire images of the road behind and on the vehicle side. The cameras are installed so that they can frame the area close to the vehicle and extend their vision over the horizon. This system can overcome some LIDAR limitations, like its inability to provide useful results when moving along dusty roads, where the clouds produced by the vehicle itself negatively affect the laser beams. This is a problem mainly of the rear LIDAR since dust clouds are produced by the vehicle in motion; sometimes it affects also the front ones especially during sudden stops when dust clouds overtake the car. Despite this problem, LIDAR data have been used to refine the detected obstacles position measurement, thus reducing the errors introduced by vehicle pitching, since distance estimation is performed by a monocular system.

- Lane Keeping: navigation in a urban environment has to be very precise. The vehicle must be able to detect obstacles and lane markings with high confidence on the detection probability and position accuracy; moreover the system must

Figure 5.2: BRAiVE external and internal equipment.

detect stop lines, to allow a precise positioning of the vehicle at intersections, and lane markings in sharp curves. The perception system must therefore cover a distance range useful for detections at driving at medium-to-high speeds. The vehicle includes two stereo systems (one on the front and one at the back) which provide precise sensing in its close proximity.

- Parking Assistance: the system aims at supporting the driver during parking manoeuvres.

- ACC Stop-and-Go: this functionality is to be used for speed adjustment in respect of the preceding vehicle. Stop-and-Go is for queue-driving, when the vehicle speed is lower than 15 km/h in a urban-like environment (whereas ACC operates at higher speeds, like on highways). Once engaged, gas and brake control are handled by the system until some driver's action on the pedals disables the automatic speed control of the car.

To meet these requirements, a sensors belt has been set up around the vehicle. The choice to primarily use passive sensors is dictated by the will of producing a car suitable for mass production: active ones are likely to interfere with each other, thus potentially degrading the performance of multiple autonomous vehicles operating in close proximity. The vehicle is equipped with a variety of devices, including sensors for world perception, navigation and control systems, localizations devices, computers, displays, batteries, wiring. A lot of effort must be spent to obtain a high integration level, making the perception devices barely visible from outside the vehicle and the controls and displays integrated into the dash board, headrest and armrest. Figure 5.2 shows the overall equipment schema. BRAiVE will be used for testing both UPDA and ADAS applications.

The work described in this thesis analyzes the driver-vehicle interaction and gives two examples of ADAS and autonomous vehicle application. Finally BRAiVE prototype has been described and its final aim is to develop and test the innovative concepts described in these chapters: from *Assisted* driving to *Fully automated* driving.

Bibliography

[1] Deborah Braid, Alberto Broggi, and Gary Schmiedel. The TerraMax Autonomous Vehicle. *Journal of Field Robotics*, 23(9):693–708, September 2006.

[2] 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.

[3] Matthias Schulze, Tapani Mäkinen, Joachim Irion, Maxime Flament, and Tanja Kessel. IP_D15: Final report. Technical report, Preventive and Active Safety Applications – Integrated Project, 2006.

[4] 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.

[5] Y.Y. Nguwi and A.Z. Kouzani. A Study on Automatic Recognition of Road Signs. In *Procs. IEEE Conference on Cybernetics and Intelligent Systems*, pages 1–6, Bangkok, Thailand, June 2006.

[6] Alberto Broggi, Pietro Cerri, Paolo Medici, Pier Paolo Porta, and Guido Ghisio. Real Time Road Signs Recognition. In *Procs. IEEE Intelligent Vehicles Symposium 2007*, pages 981–986, Istanbul, Turkey, June 2007.

[7] Dariu Gavrila. Traffic Sign Recognition Revisited. In *Mustererkennung 1999, 21. DAGM-Symposium*, pages 86–93, London, UK, 1999. Springer-Verlag.

[8] Gareth Loy and Nick Barnes. Fast Shape-based Road Signs Detection for a Driver Assistance System. In *Procs. IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems*, pages 70–75, Sendai, Japan, September 2004.

[9] S. Vitabile, G. Pollaccia, and G. Pilato. "Road signs recognition using a dynamic pixel aggregation technique in the HSV color space. In *Procs. of Intl. Conf. on Image Analysis and Processing*, pages 572–577, Palermo, Italy, September 2001.

[10] A. de la Escalera, J. M. Armignol, and M. Mata. Traffic sign recognition and analysis for intelligent vehicles. *Image and Vision Computing*, 21(3):247–258, March 2003.

[11] X. Gao, N. Shevtsova, K. Hong, S. Batty, L. Podladchikova, A. Golovan, D. Shaposhnikov, and V. Gusakova. Vision Models Based Identification of Traffic Signs. In *Procs. of Europ. Conf. on Color in Graphics Image and Vision*, pages 47–51, Poitiers,France, April 2002.

[12] Aryuanto Soetedjo and Koichi Yamada. Fast and Robust Traffic Sign Detection. In *Procs. IEEE Intl. Conf. on Systems, Man and Cybernetics 2005*, volume 2, pages 1341–1346, October 2005.

[13] W. G. Shadeed, D. I. Abu-Al-Nadi, and M. J. Mismar. Road traffic sign detection in color images. In *Procs. IEEE 10th Intl. Conf. on Electronics, Circuits and Systems, 2003*, volume 2, pages 890–893, December 2003.

[14] A. de la Escalera, L. E. Moreno, E. A. Puente, and M. A. Salichs. Neural Traffic Sign Recognition for Autonomous Vehicles. In *Procs. IEEE 20th Intl. Conf. on Industrial Electronics, Control and Instrumentation*, volume 2, pages 841–846, September 1994.

[15] Helene Hoessler, Christian Wöhler, Frank Lindner, and Ulrich Kreßel. Classifier Training based on Synthetically generated Samples. In *Procs. 5th Intl. Conf. on Computer Vision Systems*, Bielefeld, Germany, March 2007.

[16] Gang-Yi Jiang and Tae Young Choi. Robust Detection of Landmarks in Color Image Based on Fuzzy Set Theory. In *Procs. IEEE 4th Intl. Conf. on Signal Processing*, volume 2, pages 968–971, October 1998.

[17] A. de la Escalera, L. E. Moreno, M. A. Salichs, and J. M. Armignol. Road Traffic Sign Detection and Classification. *Industrial Electronics, IEEE Transactions on*, 44:848–859, December 1997.

[18] Fann Jeun-Haii and Lee Gang. A Vision-Aided Vehicle Driving System: Establishment of a Sign Finder System. In *Procs. Conf. on Vehicle Navigation and Information Systems, 1994*, pages 33–38, August–September 1994.

[19] Shashi D. Buluswar. *Color-Based Models for Outdoor Machine Vision*. Ph.D. dissertation, University of Massachusetts Amherst, 2002.

[20] Yong-Jian Zheng, W. Ritter, and R. Janssen. An Adaptive System for Traffic Sign Recognition. In *Procs. of the Intelligent Vehicles 1994 Symposium*, pages 165–170, October 1994.

[21] A. Soetedjo and K. Yamada. Fast and robust traffic sign detection. *Systems, Man and Cybernetics, 2005 IEEE International Conference on*, 2:1341–1346 Vol. 2, 10-12 Oct. 2005.

[22] N. Barnes and A. Zelinsky. Real-Time Radial Symmetry for Speed Sign Detection. In *Procs. IEEE Intelligent Vehicles Symposium, 2004*, pages 566–571, June 2004.

[23] A. de la Escalera, J. M. Armingol, and M.A. Salichs. Traffic Sign Detection for Driver Support Systems. In *3rd Intl. Conf. on Field and Service Robotics, Espoo, Finlandia, 2001*, June 2001.

[24] Y. Aoyagi and T. Asakura. A Study on Traffic Sign Recognition in Scene Image Using Genetic Algorithms and Neural Networks. In *Procs. IEEE 22nd Intl. Conf. on Industrial Electronics, Control, and Instrumentation, 1996*, volume 3, pages 1838–1843, August 1996.

[25] A. de la Escalera, J.M. Armingol, J.M. Pastor, and F.J. Rodriguez. Visual sign information extraction and identification by deformable models for intelligent vehicles. *Intelligent Transportation Systems, IEEE Transactions on*, 5(2):57–68, June 2004.

[26] S. Maldonado-Bascon, S. Lafuente-Arroyo, P. Gil-Jimenez, H. Gomez-Moreno, and F. Lopez-Ferreras. Road-sign detection and recognition based on support vector machines. *Intelligent Transportation Systems, IEEE Transactions on*, 8(2):264–278, June 2007.

[27] S. Lafuente-Arroyo, P. Gil-Jimenez, R. Maldonado-Bascon, F. Lopez-Ferreras, and S. Maldonado-Bascon. Traffic sign shape classification evaluation i: Svm using distance to borders. *Intelligent Vehicles Symposium, 2005. Proceedings. IEEE*, pages 557–562, 6-8 June 2005.

[28] P. Gil-Jimenez, H. Gomez-Moreno, P. Siegmann, S. Lafuente-Arroyo, and S. Maldonado-Bascon. Traffic sign shape classification based on support vector machines and the fft of the signature of blobs. *Intelligent Vehicles Symposium, 2007 IEEE*, pages 375–380, 13-15 June 2007.

[29] G. Loy and N. Barnes. Fast shape-based road sign detection for a driver assistance system. In *Procs. IEEE 10th Intl. Conf. on Intelligent Robots and Systems, 2004*, volume 1, pages 70–75, September–October 2004.

[30] Prealux. http://www.prealux.it/.

[31] Claudio Caraffi, Elena Cardarelli, Paolo Medici, Pier Paolo Porta, Guido Ghisio, and Gianluca Monchiero. An algorithm for Italian de-restriction Signs Detection. In *Procs. IEEE Intelligent Vehicles Symposium 2008*, pages 834–840, Eindhoven, Netherlands, June 2008.

[32] Lightweight neural network++. http://lwneuralnetplus.sourceforge.net/.

[33] Matthias Schulze. Contribution of PReVENT to the safe cars of the future. In *13th ITS World Congress*, London, UK, October 2006.

[34] Alberto Broggi, Paolo Medici, and Pier Paolo Porta. StereoBox: a Robust and Efficient Solution for Automotive Short Range Obstacle Detection. *EURASIP Journal on Embedded Systems – Special Issue on Embedded Systems for Intelligent Vehicles*, June 2007. ISSN 1687-3955.

[35] Massimo Bertozzi, Alberto Broggi, and Alessandra Fascioli. Stereo Inverse Perspective Mapping: Theory and Applications. *Image and Vision Computing Journal*, 8(16):585–590, 1998.

[36] Alberto Broggi, Stefano Cattani, Pier Paolo Porta, and Paolo Zani. A Laserscanner–Vision Fusion System Implemented on the TerraMax Autonomous Vehicle. In *Procs. IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems*, pages 111–116, Beijing, China, October 2006.

[37] Alberto Broggi, Andrea Cappalunga, Claudio Caraffi, Stefano Cattani, Stefano Ghidoni, Paolo Grisleri, Pier Paolo Porta, Matteo Posterli, Paolo Zani, and John Beck. The Passive Sensing Suite of the TerraMax Autonomous Vehicle. In *Procs. IEEE Intelligent Vehicles Symposium 2008*, pages 769–774, Eindhoven, Netherlands, June 2008.

[38] Alberto Broggi, Claudio Caraffi, Pier Paolo Porta, and Paolo Zani. The Single Frame Stereo Vision System for Reliable Obstacle Detection used during the 2005 Darpa Grand Challenge on TerraMax. In *Procs. IEEE Intl. Conf. on Intelligent Transportation Systems 2006*, pages 745–752, Toronto, Canada, September 2006.

Acknowledgements

First of all I would like to thank Alberto, for his continuous research and human support. I wanna thank also all the people that passed by VisLab and in particular the "core" of the lab for being good friends.

A special thank goes to my family for having been always present during these years. And the last but not the least, I have to mention ALL my friends that have been a precious support.