Available online at www.sciencedirect.com

ScienceDirect

IFAC-PapersOnLine 49-11 (2016) 109–114

Automatic recognition of driving scenarios for ADAS design

Alberto Lucchetti [*], **Carlo Ongini** [*], **Simone Formentin** [*],
Sergio M. Savaresi [*], **Luigi Del Re** [**]

[*] *Dipartimento di Elettronica, Informazione e Bioingegneria,*
Politecnico di Milano, via G. Ponzio, 34/5 - 20133 Milano, Italy
(e-mail: name dot surname at polimi dot it).
[**] *Institute for Design and Control of Mechatronical Systems,*
Johannes Kepler University of Linz, Altenberger Strasse, 69, A-4040
Linz, Austria (e-mail: luigi dot delre at jku dot at).

Abstract In this paper, a method to characterize and automatically recognize the most common driving scenarios in on-road experiments is presented. The aim of the proposed approach is to build a suitable simulator to develop and test Advanced Driver Assistance Systems (ADAS's). Therefore, unlike most of the existing algorithms, the whole procedure takes advantage of the intrinsic off-line nature of the problem. Context-free grammars are shown to be an effective and suitable tool for modeling the driving scenarios, while experimental results are used to validate the proposed approach and show limits and potential of a real-world application.

Keywords: ADAS, simulation, driving scenario detection

1. INTRODUCTION

In the last decade, a massive effort in the automotive field has been focused on pioneering new solutions to increase the road safety. Many feedback control schemes have been conceived to correct the driver's actions during dangerous maneuvers, e.g. in case of sudden braking, see Savaresi and Tanelli [2010]. In many other cases, the aim of the feedback systems is not to overlap the driver's will but only to be of some help with additional sensing or actuation solutions. Such systems are commonly known as *Advanced Driver Assistance Systems* (ADAS's). The decision of the European Union in 2010 to improve the road safety to reduce the number of deaths down to 20K per year (see Commission et al. [2001]) further encouraged the development of more and more effective ADAS's. A major issue related to the development of such systems is that they require an intensive testing phase on a large number of driving scenarios before being approved. In Lesemann et al. [2010], the most relevant testing scenarios for critical situations are clearly defined. The development and the test phases are usually performed in appropriate simulation environments. The selected test experiments need to be very realistic to mimic the real driver's behaviour and the surrounding environment. Moreover, it is important to extract the driver's reactions in the same situations considering different countries/cultures, see Özkan et al. [2006]. To this aim, an instrumented vehicle can be used to record and detect the most significant scenarios and analyze their dynamics.

In this paper, we present a general procedure for the classification and automatic recognition of the most common driving scenarios, which can be observed in the experimental tests, with the aim to mimic them in a simulation environment.

In the scientific literature, many procedures have been already proposed for the detection and the classification of such scenarios during driving experience. As far as we are aware, the existing studies have been focused only on a subset of the driving scenarios treated here (see, e.g. Fastenmeier and Gstalter [2007]). Moreover, the existing algorithms have been developed with the aim of an on-line detection, and not for off-line description. This fact makes a big difference in that some scenarios can be more accurately described and more easily detectable off-line (because the entire trajectories of the vehicles are known), as we will show in the paper. In Ryoo et al. [2013], a tool for the detection of critical driving situations has been developed with the purpose of classifying the drivers' behaviour. In Wang et al. [2005], the vehicle behaviour is classified as normal or abnormal. In other contributions, a specific scenario is analyzed, e.g. overtaking in Ramirez et al. [2014], turning maneuvers in Barth and Franke [2010] and lane changes in Kasper et al. [2012].

To classify the driving scenarios, we follow a *divide et impera* approach. We define simple events based only on the relative position among the vehicles, the *atomic events*. Then, we apply *context-free grammars*, to define a "language" describing more complex events. Context-free grammars are information-theoretic tools usually employed to develop compilers and parsers for a given language and therefore they are very suited to verify whether a prespecified event has occurred (i.e. a sequence of atomic events is accepted by the language). As far as we are aware, this is the first contribution where such tools are used in this framework.

The paper is organized as follows. In Section 2 we present in detail the experimental setup. In Section 3, we discuss the data pre-processing needed for our goal. The scenarios

10.1016/j.ifacol.2016.08.017

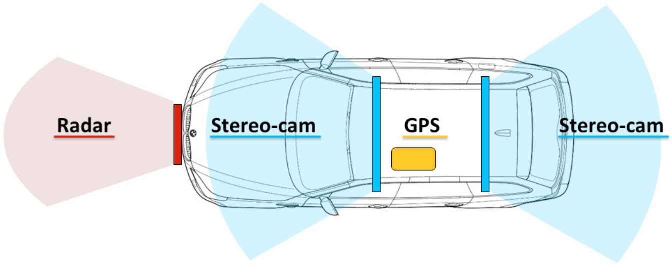

Figure 1. Experimental setup.

	XB3	UI-1221LE
Frame rate [Fps]	16	87.2
Resolution [MPix]	1.23	0.36
Resolution [h x v]	1280x960	752x480
Shutter	Global	Global
Interface	2x9-pin IEEE-1394b	USB 2.0

Table 1. Front (XB3) and rear (UI-1221LE) cameras.

detection algorithm is presented in Section 5 following the taxonomy in Section 4. Section 6 shows the results and the validation of the procedure. We end the paper with some concluding remarks.

2. SYSTEM DESCRIPTION AND SET-UP

The experimental setup is illustrated in Figure 1. The vehicle, from now on indicated as $\mathcal{V}$, is equipped with the following sensors:

- a radar sensor in the frontal position;
- a stereo-camera in the rear position;
- a dSPACE AutoBox system for acquiring the CAN BUS vehicle at $10Hz$.

A frontal stereo-camera will be also used for validating the scenario detection algorithm using the radar sensor.

The radar sensor has a distance range from $0.25m$ up to $200m$ with a field of view of $\pm 8.5°$. The resolution and accuracy distance are $2m$ and $0.25m$ respectively, while the resolution and accuracy angle are $1°$ and $0.1°$ respectively. This radar can track 40 objects at the same time.

The features of the stereo cameras are reported in Table 1. We acquire the images on two different laptops with a frame-rate of $10fps$, which is the same sample time of the dSPACE system.

3. DATA PRE-PROCESSING

Before detecting the driving scenarios, a data processing procedure is needed.

First of all, notice that most of the objects that we acquire with the radar are not useful for our purposes, e.g. walls, road signs or parked cars. In Figure 2 a sample of an acquisition made by the radar sensor is shown. In the subplots are illustrated, from top to bottom: the speed of $\mathcal{V}$, the longitudinal displacement and the velocity of the tracked objects. It is clear from the second subplot that most of the objects are not useful and the third subplot highlights that for most of them the absolute velocity is close to zero. For our aim, such quasi-static objects can be deleted, thus decreasing the computational load of the overall algorithm. The resulting plot is illustrated in Figure 3.

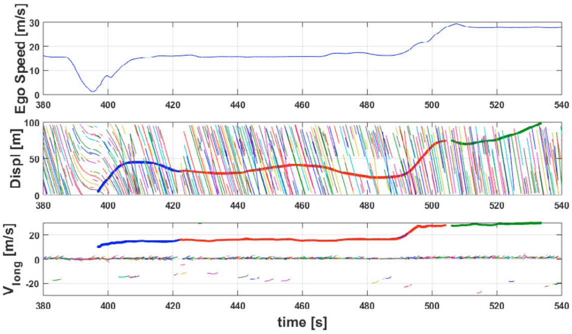

Figure 2. Radar acquisition sample: raw data.

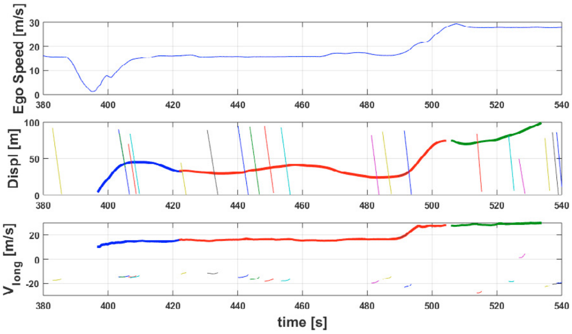

Figure 3. Radar acquisition sample: processed data.

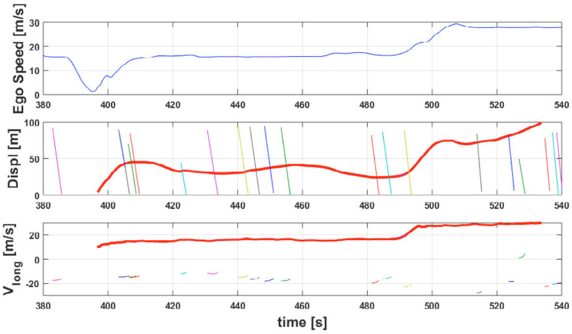

Figure 4. Associated measurements.

Secondly, to prevent any object tracking loss, first we check if any object has been lost within the field of view of the radar or cameras without reaching the boundary. If such an object exists, we look for other objects that appear within the field of view. These objects are candidate to be the same vehicle that has disappeared. Then we compare the predicted trajectory (obtained considering constant speed) with that of the candidates, we select the one that minimizes the prediction error and we associate the two vehicles. In Figure 4 the results of such a procedure is illustrated for a case where the vehicle tracked at time $395s$ is associated with the one tracked at time $425s$ and one at $507s$.

The final step of the pre-processing phase is to characterize the position of the other vehicles with respect to $\mathcal{V}$. This is not straightforward, as in the characterization of