



# On-board monitoring system for road traffic safety analysis

Sebastiano Battiato, Giovanni Maria Farinella, Giovanni Gallo, Oliver Giudice\*

Image Processing Laboratory, Dept. of Mathematics and Computer Science, University of Catania, Italy



## ARTICLE INFO

### Article history:

Received 31 August 2017

Received in revised form 20 December 2017

Accepted 25 February 2018

Available online xxx

### Keywords:

On-board systems

Traffic safety

Stereo vision

## ABSTRACT

This paper presents a framework for road traffic safety analysis. It is based on a stereo-vision system that, after being installed on-board of public transportation vehicles, collects data of what happens in front of a moving vehicle. The collected data are analysed throughout a process that acquire raw GPS information, video sequences and stereo-based depth maps to compute the surrogate safety measures. These measures are obtained by exploiting the Traffic Conflict Technique in conjunction with computer vision algorithms and a cascade of classifiers. The safety measures are then used for further analysis in order to identify dangerous locations in which an intervention is needed to improve the safety level and prevent accidents. Experiments performed in a real urban environment confirm the effectiveness of the framework.

© 2018 Elsevier B.V. All rights reserved.

## 1. Introduction

Road accidents are one of the leading causes of death, therefore road safety is a crucial and a delicate matter for national governments to be solved. Over the last years, governments have launched a series of initiatives with the intention to significantly reduce road accidents by acting directly on the three basic interacting elements of the road: the human behaviour, the vehicles security features and the road infrastructure. Human behaviour is the most difficult to deal with, because it can be predicted only to a limited extent. The improvements in vehicle safety are dependent on the implementation of new technologies and security features by the manufacturers. Finally the road infrastructure has a key role in safety influencing both human behaviour and vehicles performances. Relevant road infrastructure factors include the quality of the road illumination, the presence and the readability of signs and road markings as well as the quality of the paving [1]. Road safety is hence the final outcome of regulations and prevention techniques that, acting on the above mentioned three basic elements, make roads more secure. A method to objectively measure the safety level of a road may help to choose the most effective intervention. The research for such a measure motivated many studies. All of these can be divided into two main categories: those based on the analysis of statistical data of accidents [1] and those based on road users analysis and inter-

operation, called naturalistic studies [2]. Most of the statistical studies are performed by evaluating crash data collected over a long period of time. They focus on rigorous statistical models and can tell something about the level of safety only after a certain number of events has already occurred. However the naturalistic studies can find the causes that can lead to an accident in order to prevent it. The actual difficulty in naturalistic studies is to collect and access the data needed for a good output. Current technologies offer the opportunity to collect data continuously with networks of cameras, wireless sensor networks and on-board monitoring systems. The framework presented in this paper can automatically collect data, while moving on board of a vehicle, and locate dangerous places for road users in order to send alarms to experts and allow a prompt intervention to improve the safety level of the location. To this aim, the framework is composed of a video-based on-board monitoring system that exploits the model of the Traffic Conflict Technique (TCT) to compute safety measures in conjunction with computer vision techniques and classification algorithms. This framework mounted properly on a public bus transportation system, which runs every day the same roads, can create a distributed moving sensor network which continuously collects data directly on the field. Compared to other solutions of the state of the art, the proposed approach provides more information relative to pre-crash and crash events than what is currently available, it is less expensive. Moreover it allows to investigate with few sensors a very large territory and it has safety risk detection with real-time capabilities. Finally it is not based on the analysis of the driver behaviour, but it focuses only on the interaction between road users, making it less intrusive with respect to privacy issues.

\* Corresponding author.

E-mail addresses: [battiato@dmi.unict.it](mailto:battiato@dmi.unict.it) (S. Battiato), [gfarinella@dmi.unict.it](mailto:gfarinella@dmi.unict.it) (G.M. Farinella), [gallo@dmi.unict.it](mailto:gallo@dmi.unict.it) (G. Gallo), [giudice@dmi.unict.it](mailto:giudice@dmi.unict.it) (O. Giudice).

The paper is organised as follows. The state of the art techniques related to traffic analysis approaches are discussed in Section 2. We then introduce the reader to the Traffic Conflict Technique fundamentals in Section 3. The proposed framework is described in Section 4. Section 5 discusses the experimental results obtained considering real data collected in the city of Catania – Italy. Finally, conclusion and directions for future works are given in Section 6.

## 2. Related work

Efforts to reduce risks and improve safety on roads are made through many strategies that typically involve processes of driver screening and selection, driver training, vehicle maintenance and road safety inspections in addition to the statistical analysis of accident data.

While those approaches have had some success, they are slow and very expensive and often they fail to have a sufficiently broad and reactive view of the reality in order to swiftly identify problems for road safety and thus to prevent accidents [1]. Naturalistic studies have demonstrated to be more reactive in identifying problems for safety [2] and nowadays technologies offer the opportunity better results with systems that can detect problems with safety and automatically trigger an intervention to solve it. It is possible to differentiate between those systems based on static positioned sensors (single or multiple) and those based on sensors mounted on board of vehicles.

In [3,4] many techniques were presented, mostly focused on driver behaviour analysis. Their studies give the inspiration for a possible broad classification of all analysis techniques into: Real-Time techniques, able to identify a safety issue at current time and non Real-Time ones, able to collect and analyse data from numerous samples.

Based on the devices exploited, it is possible to further classify the state of the art techniques into: fixed-position sensor networks and on-board systems.

There is a huge number of studies that use networks of sensors placed in critical positions, specifically to study the traffic flow or the vehicles behaviour [5–8]. These studies use various techniques to collect and analyse video data exploiting the TTC model [6–8] in order to identify pre-crash situations. They are very reliable and have a limited cost. However the positioning factor is crucial because these systems allow you to monitor only areas that already are thought to be dangerous therefore do not help to identify high-risk areas which are not known. In addition roads and the traffic flows constantly change, so fixed-placed solutions need to be constantly upgraded and re-positioned as well.

On-board systems allow to know what happens around the vehicle while moving together with it. Most systems are based on the sensors of the vehicle itself [9] (speed, acceleration, GPS, etc.), while others [10] are video-based and can achieve great accuracy in identifying drivers behaviour. Horrey et al. [11] presented a detailed review of the state of the art of on-board monitoring systems. However, despite they have good accuracy in identifying driver's behaviour their solution lacks in detecting locations where road safety is compromised.

As regards on-board systems, mobile-phone based solutions are promising in terms of being able to collect a very big amount of data exploiting the widespread of mobile devices among people. Among others in [12] mobile phone sensors (GPS receiver and orientation sensor) were used to detect erratic driving behaviour caused by overtaking considering phone as a steering wheel. Another interesting study that uses the three-axis accelerometer sensors of smart-phones was developed in [13]. The system presented by the authors claims to be used to detect and analyse various external road conditions and driver behaviours that could be dangerous to the health of the driver, automobile and the

neighbouring public. In [14] the authors proposed a vehicle mounted system which provides a safety index to each driver on the basis of his quality of driving. The android device was fixed to the windshield and captured the audio and video signals (10 s before and after the event) and in order to detect driving manoeuvres like aggressive turn and sudden brake etc., thresholds were defined for the accelerometer in each direction. A similar study was developed in [15] where the authors presented a mobile application that combines the sensors of mobile phone like GPS, accelerometer and microphone to detect the driver behaviour, traffic and road conditions.

While promising mobile-based studies work in extremely unconstrained conditions and with a various quantity of low-quality sensors. Thus the overall quality of collected data is extremely low. Moreover, given the big amount of data, the analysis and insight extraction should be done offline.

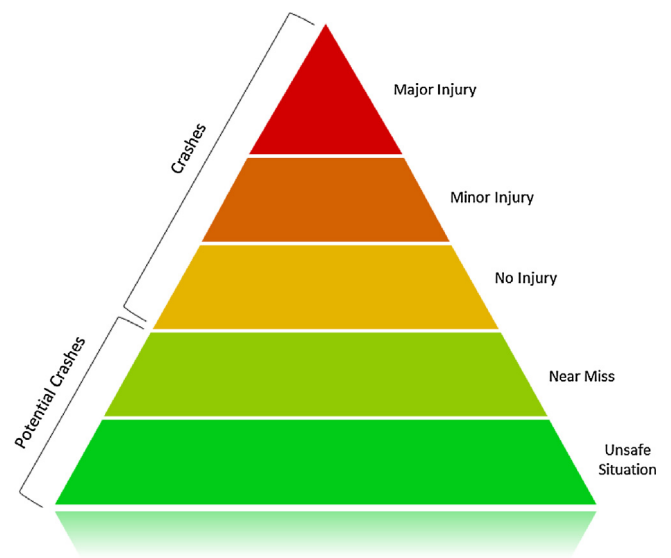
Studies conducted in [16] demonstrated the effectiveness of using an on-board monitoring system based on stereo-vision to automatically compute measures useful to compute a level of risk in a detected pre-crash event.

As discussed in [17], the research and development of on-board monitoring systems should ideally: (a) identify and validate behaviours that may be precursors to crashes or injuries; (b) be based on cost-effective ways to monitor behaviour; (c) establish management and driver acceptance of the program.

In this paper we present the implementation of a framework for road safety analysis that satisfies all of the above mentioned conditions and fills the gap in the field of automatic road safety analysis combining high-quality real-time video and sensor data collection and analysis techniques in a mixture of real-time and non real-time analysis.

## 3. The Traffic Conflicts Technique

The 'Heinrich Triangle' theory [18] provides a conceptual framework to reason about road accidents (Fig. 1). It is founded on the relationship that 'no-injury accidents' precede 'minor injuries' (i.e., events closer to the base of the triangle precede events nearer the top). Moreover the 'Heinrich Triangle' events near the base occur more frequently than events near the top.



**Fig. 1.** The Heinrich Triangle describes crash events in terms of injury severity and near crash or potential crash events in terms of risk. Events at the base of the triangle happen more frequently.

The Traffic Conflict Technique (TCT) takes off from the ‘Heinrich Triangle’ theory, assuming that the appropriate Traffic Conflict (TC) factors can be defined as measures of near-crash events. A TC is defined as an observable situation in which two or more road users approach each other in space and time to such an extent that there is risk of collision if their behaviour remain unchanged [19] (e.g. a pedestrian crossing the road and a vehicle approaching that does not brake or change direction).

The TCT is based on the measurement of both spatial and temporal variables describing the interactions between two road users (e.g., a car and a pedestrian) involved in a critical event for safety. The Time To Collision (TTC) is the main indicator used by the Traffic Conflict Technique and it is computed for the vehicles (TTCv) and obstacles (TTCo). Formally TTC is defined as follows [20,16]:

$$TTC = \frac{D}{V} \tag{1}$$

where  $D$  is the distance between the subject and the conflict point and  $V$  is the velocity of the road users (the vehicle or the generic obstacle).

Considering the event dynamics it is possible to differentiate the Traffic Conflicts into three major classes with related TTC measures (Fig. 2: *Pedestrian Conflict, Car Following Conflict and Intersection Conflict*).

In case the obstacle is a pedestrian crossing the road we have a *Pedestrian Conflict*. The TTC measures, related to this conflict, at

time  $i$  for the two involved actors (i.e., vehicle and pedestrian), are defined as follows:

$$TTC_i(v) = \frac{D_{z_i}(v)}{V_i(v)}; \tag{2}$$

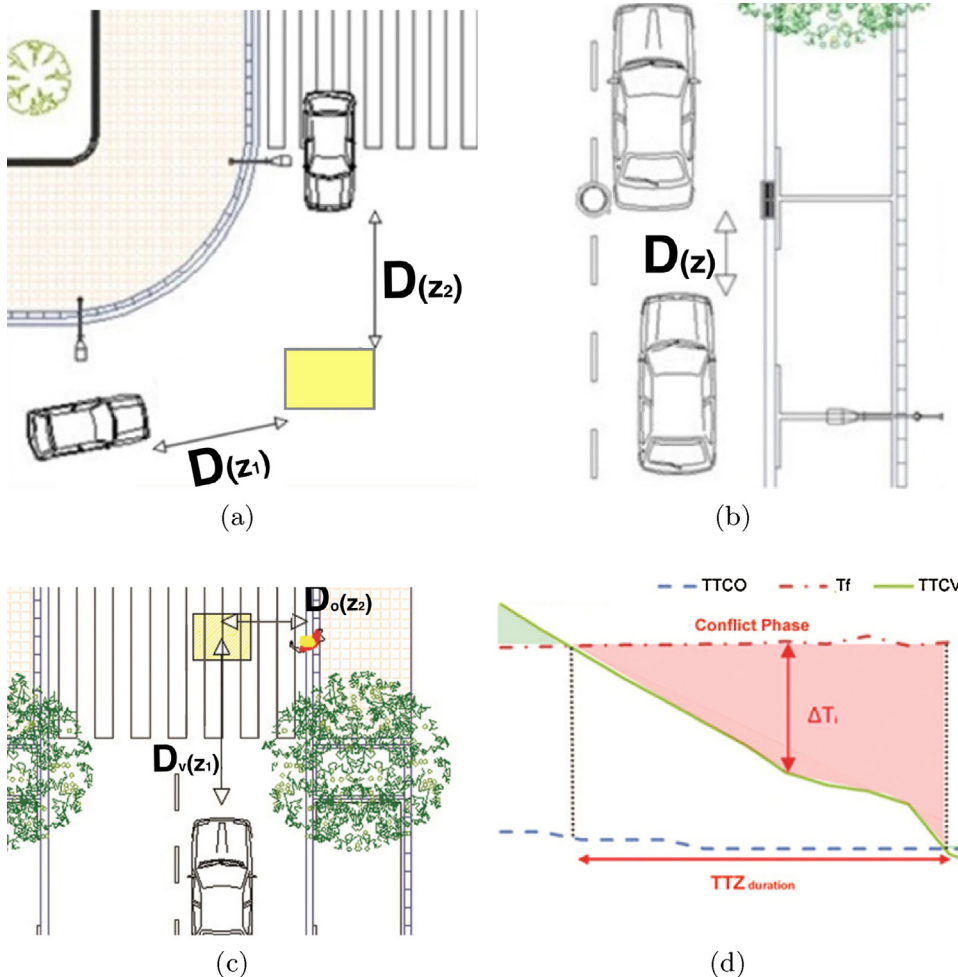
$$TTC_i(p) = \frac{D_{x_i}(v) - D_{x_i}(p)}{V_i(p)}. \tag{3}$$

where by considering a 3D reference system centred on the vehicle,  $D_{z_i}(v)$  represents the distance between the vehicle and the pedestrian along the  $Z$  axis and  $D_{x_i}(v) - D_{x_i}(p)$  represents the distance between the vehicle and the pedestrian along the  $X$  axis (see Fig. 2c). Let be  $T_{f_i}$  the time required for the vehicle to stop at time  $i$  defined as:

$$T_{f_i} = T_r + \frac{V_i(v)}{a_f} \tag{4}$$

where  $T_r$  is the reaction time,  $V_i(v)$  is the vehicle velocity at time  $i$ ,  $a_f$  is the deceleration during braking. Taking into account the previous considerations it is possible to differentiate the following cases, which define the areas marked in Fig. 2d:

- $TTC_i(v) > T_{f_i}$ : vehicle may stop before conflict area,
- $TTC_i(p) > TTC_i(v)$ : the vehicle passes the area of conflict before the pedestrian reaches it,



**Fig. 2.** The three classes of TC: (a) Intersection, (b) Car Following, (c) Pedestrian Crossing, (d) Temporal trend of the quantities involved in the TTC. The area where  $TTC_o(i) < TTC_v(i) < T_f(i)$  is related to the conflict. Each actor has a reference system integral to itself.

- $TTC_i(p) < TTC_i(v) < Tf_i$ : *Pedestrian Conflict*.

If the conflict exists it is possible to evaluate its duration ( $TTZ_{duration}$ ) as the difference between the final and initial time ( $i_f - i_0$ ) in which Traffic Conflict conditions are satisfied.

The second traffic conflict class is the *Car Following* (Fig. 2b) in which there are two vehicles moving along the same direction. The conflict occurs when the vehicle in front ( $v_1$ ) of another vehicle ( $v_2$ ) makes a rush braking or a fast deceleration. If the distance to actually stop for vehicle  $v_1$  is equal or minor to the distance of the vehicle  $v_2$  in front of it, the crash will occur. The TTC measure at time  $i$  for the second vehicle  $v_2$  can be defined as stated in [21]:

$$TTC_i = \frac{V_i^2(v_1)}{2V_i(v_2)a_i(v_1)} - \frac{V_i(v_2)}{2a_i(v_2)} + \frac{D_{y_i}}{V_i(v_2)} \quad (5)$$

where  $D_{y_i}$  is the distance between the vehicles along the Y axis at time  $i$ ,  $V_i(v_1)$  and  $a_i(v_1)$  are respectively the velocity and deceleration of the front vehicle at time  $i$ ,  $V_i(v_2)$  and  $a_i(v_2)$  are respectively the velocity and deceleration of the rear vehicle at time  $i$ .

If  $TTC_i < Tf_i$ , with  $Tf_i$  defined as in Eq. (4), the vehicle  $V_2$  could not stop in time, the conflict occurs and it is possible to evaluate  $TTZ_{duration}$ .

The third traffic conflict class is the *Intersection Conflict* (Fig. 2a). This class is very similar to the Pedestrian Conflict one assuming that the involved actors are now two vehicles. The difference is that in this class much greater speeds are involved. For this reason it is mandatory to take into account the  $Tf_i$  of both vehicles from this assumption the TTC measures of both vehicles are defined as:

$$TTC_i(v_1) = \frac{D_{y_i}(v_1)}{V_i(v_1)}; \quad (6)$$

$$TTC_i(v_2) = \frac{D_{x_i}(v_2)}{V_i(v_2)}. \quad (7)$$

where by considering a 3D reference system centred on the considered vehicle  $v_1$ :  $D_{y_i}(v_1)$  represents the distance between the first vehicle and the second along the Y axis,  $D_{x_i}(v_2)$  represents the distance between the second vehicle and the first along the X axis,  $V_i(v_1)$  and  $V_i(v_2)$  represents respectively the velocities of the first vehicle and the second one along their direction of travel.

If both conditions  $TTC_i(v_1) < Tf_i(v_1)$  and  $TTC_i(v_2) < Tf_i(v_2)$  happen, the conflict occurs and it is possible to evaluate  $TTZ_{duration}$ .

Given the  $TTZ_{duration}$  it is possible to compute the severity of the conflict as Risk Impact (RI) at time  $i$  according to the following equation:

$$RI_i = \sum_{i \in TTZ_{duration}} V_i^2 \times \Delta T_i \quad (8)$$

where  $V_i$  is the velocity of the vehicle at time  $i$  and  $\Delta T_i = Tf_i - TTC_i(v_i)$ .

It is also possible to evaluate the RI of the whole conflict as follows (according to [16]):

$$RI = \frac{\frac{1}{N} \sum_i RI_i}{TTZ_{duration}} \quad (9)$$

The discrimination between the three classes of conflict can be done automatically. The *Car Following* case is easily distinguishable from the others by analysing the magnitude and direction of speeds of both vehicles. On the other hand to discriminate between intersection and pedestrian classes the TTC measures alone are not

sufficient and visual information is needed. Automatic classification of TC classes will be introduced in Section 4.5.

#### 4. Framework description

The proposed framework is composed of several modules that operate in different phases. For clarity we first describe the hardware components in Section 4.1 whereas the software modules that run on top of them in the subsequent sections.

##### 4.1. Hardware components

The hardware employed in the proposed framework is composed of a number sensors, processors and network components that are connected together as described in Fig. 3.

The TYZX DeepSea G3 Embedded Vision System [22] (EVS in Fig. 3) is an embedded stereo image processor which is used to compute the depth-map of the scene in front of the vehicle. The design of the EVS is based on an architecture that implements the Census stereo vision algorithm [23]. As the input pixels enter the EVS, the Census transform is computed at each pixel based on the local neighbourhood, resulting in a stream of Census bit vectors. For each pixel of one image of the stereo system, a summed Hamming distance is used to compare the Census vectors around the pixel to those at the window locations in the other image. These comparisons are pipelined and occur simultaneously. The best match, shortest summed Hamming distance, is located with five bits of sub-pixel precision. The EVS Processor converts the computed pixel disparity to metric distance measurements using the stereo camera calibration parameters and the depth units specified by the user. To have a system able to compute the distance of an object from the driver perspective, we needed to take into account the working ranges and the HFOV of the EVS. We used a baseline of 33 cm and an 83 degrees HFOV lens in our experiments. This configuration has a working distance range between 2.5 and 50 m. The specifications of the EVS used in our experiments are described in Table 1.

Fig. 3 shows how the EVS is connected with a router to a GPS module and a notebook. The router solves the EVS data access problem. A gigabit Ethernet WLAN router manages a network in which the EVS is accessed by a notebook. A client can access the EVS by connecting to a TCP service that sends every image and the computed depth data. Since the required stereo videos must be geo-referenced we also connected the EVS to a Bluetooth GPS receiver module. The choice of a wireless device is mandatory to solve the signal satellite problem, since we have used a city bus for our experiments. The shape of a city bus forced us to put the GPS module on the vehicle roof to have the best signal quality. Finally a notebook PC connected to the LAN acts as data collector. The

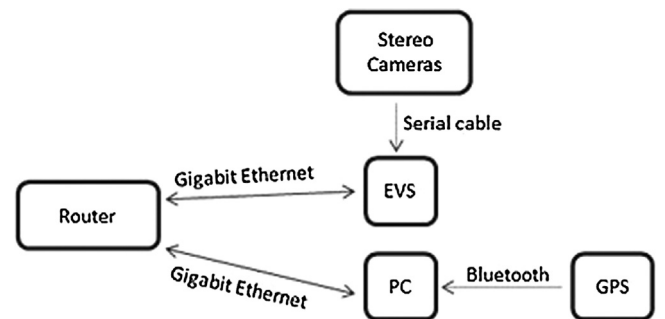


Fig. 3. The connection scheme of all the sensors that acquire raw data during the input phase.



**Table 1**  
Hardware specifications of the EVS.

G3 Embedded Vision System Specifications	
Weight	675 g
Size	3.8 cm × 18.7 cm × 14.5 cm
Power	11 W typ.: 12 vdc or PoE class III
Imagers	Aptina MT9V022 color, up to 60 fps, 752 × 480 pixels
Lens	83 degrees horizontal FOV
Stereo baseline	33 cm
CPU	Freescale PowerPC 8347 @400MHz
Memory	256 Mbytes
Operating system	Linux 2.6 Kernel

notebook controls the EVS and receives data both from the EVS and the GPS module. All further processing is done on the notebook. The overall cost of the hardware components described above is less than 2000\$ where the most expensive components are the stereo camera and the notebook. The correct installation and positioning of the camera on the vehicle is crucial for the proper functioning of the system. Camera positioning is a relevant issue because it must work without any occlusion in the field of view and should not be intrusive for the driver. In Fig. 4a a correct positioning of the camera in a bus is shown: this position is integral with the field of view of the driver and not intrusive for him.

After installation, a calibration procedure is needed. The calibration procedure consists of placing outside and in front of the vehicle two targets orthogonal to the axis of the vehicle at 3 m away from the camera. The targets are placed using a high precision laser rangefinder positioned on vehicles vertices and pointed to the targets. At this point, by using the EVS it is possible to evaluate the distances of the two targets and to record the distance difference between the right target and the left one. The estimated differences are dependent to the inclination of the camera relative to the axis of the vehicle. These values are stored and used to completely specify the reference system of the camera. Fig. 4a illustrates the position of the camera on a bus. Fig. 4b shows the system used for data acquisition in our experiments.

#### 4.2. Software modules

The software of the proposed framework analyses all data provided by hardware sensors in order to compute the TCT measures and detect locations at risk for safety. For an easier understanding we will describe the overall process divided into logical phases as shown in Fig. 5.

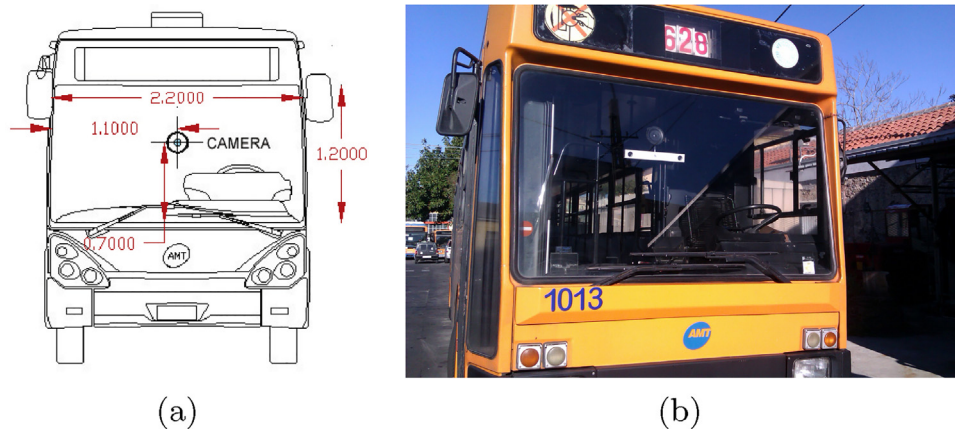
The whole process starts by gathering data from the sensors. The data collected for further processing are: the GPS coordinates, the RGB video frames and a depth map of the scene in front of the vehicle. To achieve a geo-referenced video with depth information the raw data must be synchronised in a single record to represent all the raw input variables together. The synchronization is performed in the *Data Acquisition Phase*. The data acquired in the first phase are typically too voluminous for an efficient evaluation. The successive step is hence the *Summarisation Phase* to find those data sequences eligible to identify a TC. The data sequences are then furtherly analysed to discriminate if they describe a real TC or not. Once a conflict is detected the next phase computes an estimate of the risk of an accident (*Risk Analysis Phase*). At this point the classified TC event, with the corresponding computed risk, is recorded and located on a map for further data analysis and decision making (e.g., acting on the infrastructure of the road).

The above steps are discussed in more detail in the next sections.

The presented software modules were developed in C/C++ programming language by means of QT IDE and exploiting open-source libraries and modules that will be indicated in each section.

#### 4.3. Data Acquisition Phase

The data sequences composed of colour images, depth maps and GPS data taken during the input phase, are compressed and sent to a notebook for further computation. The colour images are compressed according to the MPEG4V3. The depth-map is a 24-bit RGB image where the most significant 16 bits represent a metric distance value. These images are compiled into a video file on which we operate a lossless compression using the FFv1 [24]. The images and the corresponding depth-maps from the EVS need to be geo-referenced. The data acquisition module solves the problem of the frequency difference between the two streams. As a matter of fact image and depth data have a frequency of 20–30 Hz while the GPS data comes at 10 Hz. The acquisition module operates in two phases: when a new video information arrives the software operates a sample and hold mechanism on GPS information, so there will be some frames referring to the same GPS information. As a new GPS information arrives, the system changes previous held information by linear-interpolating the values of position and velocity according to the pattern described by the new GPS information. Finally, the software produces a text output that allows to uniquely bind each frame and depth-map to the



**Fig. 4.** The proposed system installed on an urban bus for data collection and Traffic Conflict Analysis testing. (a) Placement of the camera on a bus. (b) Photo of the installed camera.

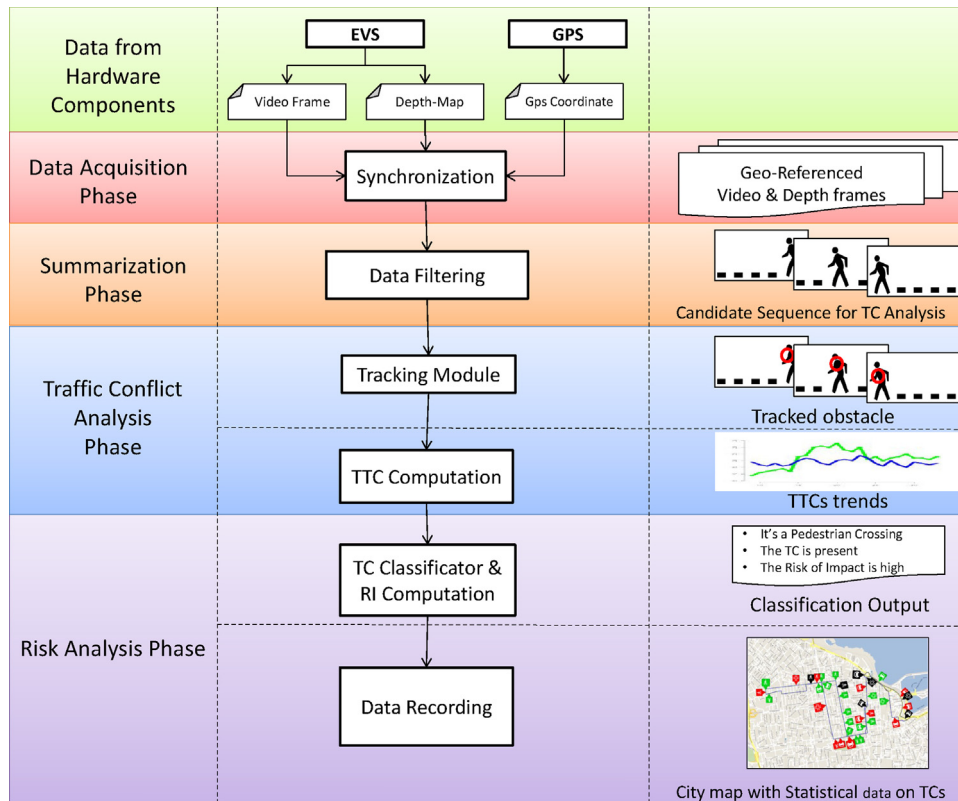


Fig. 5. The framework pipeline from input phase of raw data to a geo-referenced output of processed data.

corresponding GPS information. When cold start is performed, even with the considerations explained above, GPS and video data cannot be synchronised due to network latencies. In this case synchronisation is achieved by the acquisition module through the evaluation of these latencies on packet transmission. This was done by means of the Linux-based Network Time Protocol (NTP [25]) and the NTP C++ modules.

4.4. Summarisation Phase

Data acquisition records a number of data greater than what is really needed for our purposes. A way to reduce the amount of data produced is to make a summarisation based on the identification of video sequences that are most likely candidate to be a conflict. It has been already demonstrated that it is possible to identify these points simply by analysing only GPS tracks and measures derived from it (e.g. speed, trajectory, acceleration, jerk, angular deviation rate, etc.) in conjunction with the application of filtering techniques (based on Butterworth filter) and outlier detection

techniques (based on Mahalanobis distance). The mathematical details for the summarisation technique employed in our framework has been presented in [26] while the library used for implementation was the DLIB C++ library [27]. As output there are the initial points and the duration of each sequence in which a possible TC occur. Given the synchronization between GPS data and video, the corresponding video sequences are then taken into account as candidates for further analysis. The numerous presence of false positives however asks for a further analysis described in the next Subsection.

4.5. Traffic Conflict Analysis phase

In this phase the analysis of video data starts to take place. All video and image computation modules were developed using the OpenCV open-source project [28] (version 2.7). Once video sequences of candidate TC are detected, in previous phase, the TC analysis module proceeds to identify the obstacle that may have generated the TC. This obstacle will be represented by the point

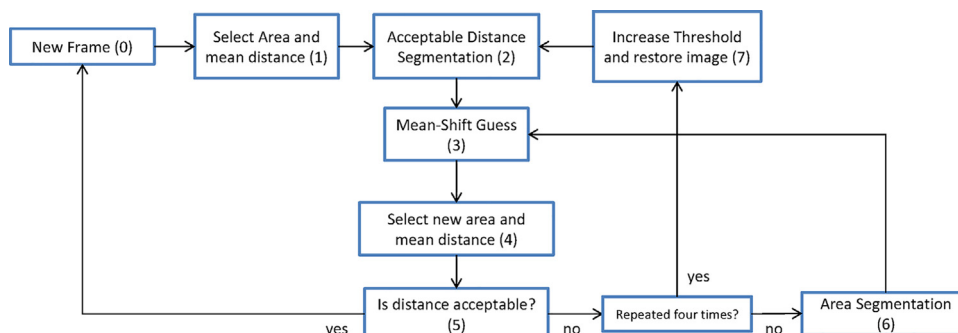


Fig. 6. Diagram of the depth based mean-shift tracking algorithm.

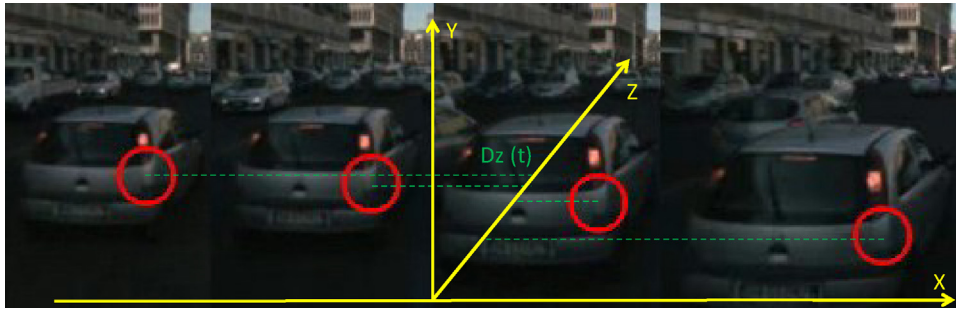


Fig. 7. TC analysis example of a candidate sequence. Four key frames of the obstacle that generated the conflict, the circle is the output of the tracking algorithm, the reference system used is shown.

that is the closest to the vehicle in the depth map. The obstacle is then considered as an object if that appears to be “big” in terms of dimensions (at least 60 cm height and 10cm width): this can be easily done by using the depth information. After the obstacle is considered as an object. The initial point of interest will be the one with the lowest value in the depth-map between the points of the detected object. This initial point will be automatically followed in subsequent frames thanks to the tracking module. The tracking module in this way produces measurements of the TC indicators described in Section 3. In our experiments an augmented version of a well-known tracking algorithm called mean-shift [29] has been developed. The mean-shift tracking algorithm has been extended to work with depth information and a-priori knowledge based on vehicle speed. To do this our development took place from the OpenCV [28] mean-shift implementation. The depth based mean-shift tracking algorithm starts from the idea that, given the vehicle speed, an obstacle in front of the vehicle cannot change in its distance value more than a certain threshold between two consecutive frames. This algorithm works as indicated by the diagram in Fig. 6. As a new frame arrives, the tracking module considers a circular patch around the initial point of interest (with radius of 10 pixels) and computes the mean distance value for all pixels in that circle taking corresponding values from the depth map (1). When the next frame arrives, for each pixel, if the difference of distance values, between the pixel in the current frame and the relative pixel from previous frame, surpasses a certain threshold, the pixel value is set to 0; otherwise is left unchanged (2). At this point, the mean-shift algorithm guesses where the new point of interest is (3) and the mean between

distance values of the pixels inside the patch around that new point of interest is computed (4). If this last mean distance is acceptable (5), with respect to the vehicle speed, then the guessed point of interest is taken as the new one, otherwise all pixels in the patch around the guessed point of interest are set to 0 and all pixels in the coordinates of the previous frame patch are restored to their original value (6). At this point mean-shift guessing is repeated. If the algorithm does not find a suitable new point of interest in four iterations, the distance threshold is increased by one meter, the image is restored and everything starts back from (2). Four iterations were needed to set to 0 four different guessing directions of mean-shift in case on inability to find an acceptable new point of interest. An exit condition for the algorithm should be selected in case of disappearance of the object. We empirically selected an initial threshold of 1 m and an exit condition of 10 m.

Using this tracking algorithm it is possible to follow an obstacle in the entire sequence of a candidate TC video sequence. Thanks to the data coming from the depth-map is possible to fully reconstruct what happened between the vehicle and the obstacle by computing relative speeds and distances of both vehicle and obstacle in X and Y axis components according to the reference system described in Fig. 7.

The obtained measurements are then used to compute the TC indicators. In Fig. 9 the indicators for the candidate TCs shown in Fig. 8 are plotted.

Starting from in depth analysis and results obtained in previous works [16] in this paper we present an improvement in terms of automatizing the TC analysis phase: through the exploitation of

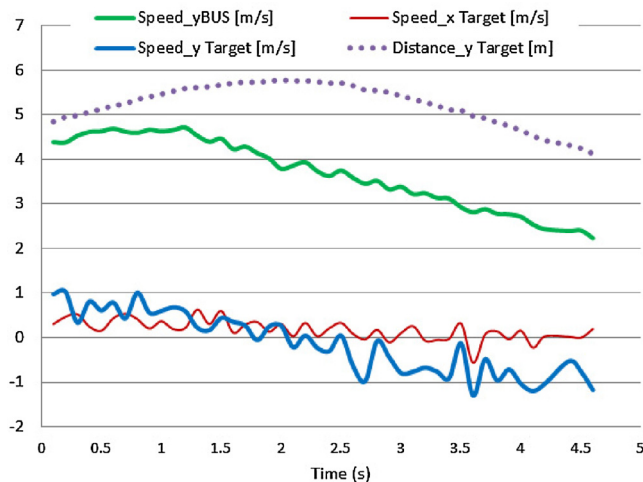


Fig. 8. Trends of speed and distance between the vehicle and the obstacle during the analysed conflict phase.

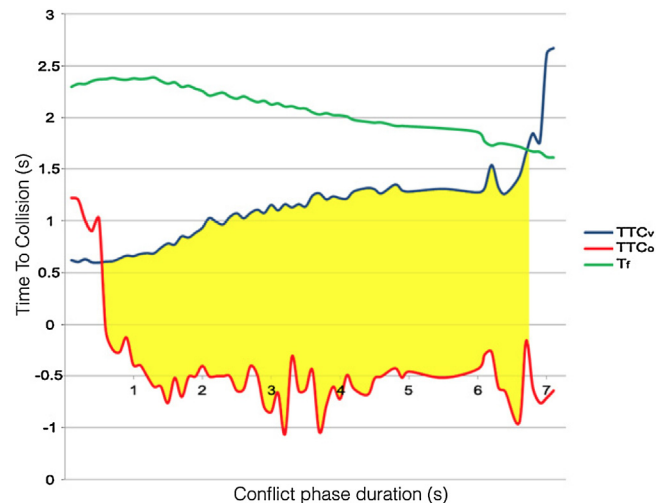


Fig. 9. TTC measures over time, the conflict is actually present and the conflict phase is highlighted in yellow. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of the article.)

the TTC values of both vehicle and obstacle, and using image analysis techniques it is possible to setup a cascade of classifiers that can automatically infer if a conflict is actually present and to classify it into one of the three classes introduced in Section 3. The first classifier (developed using the DLIB C++ Library [27]) discriminates between two kind of TCs: one in which the obstacle is moving in a direction perpendicular to the moving direction of the vehicle (this class represents both pedestrian and intersection conflicts) and a second one in which the obstacle is moving in the same direction of the vehicle (this class is only represented by the car following TC). Given this distinction, it is easy to discriminate between these two groups of TCs by analysing the speeds of the vehicle and the obstacle with respect to their components. Through linear regression on the speed values, the coefficients of the lines thus obtained are used as features for a Bayesian classifier. The classifier is trained with a set of data manually labelled by an expert on an acquisition of 1 h in which 100 sequences of TC candidates were identified.

A second classifier is needed to distinguish the pedestrian from the intersection case. To this aim the HOG based pedestrian detector [30] (implemented in OpenCV [28]) was used. The detector gives information on the presence of the pedestrian: if there is a pedestrian it should be placed moving in front of the vehicle and with horizontal moving direction. In our experiments, the pedestrian detector performed better than expected given the input of the “big enough” obstacle detected earlier. Finally a third classifier determines the existence of the real TC by using conditions described in Section 3. The final output of the Traffic Conflict Analysis phase is a decision on a candidate TC in order to identify the type of conflict and if it is a real conflict. The Risk Analysis Phase is done only if candidate TC is classified as positive conflict.

#### 4.6. Risk Analysis Phase

Once a TC is identified it is possible to compute the RI according to the formulas described in Section 3. The RI is normalised on the duration of the conflict and represented as one between three levels: low (lower than 0.50), medium (between 0.50 and 0.75) and high risk (over 0.75). The values that characterise these risk factors have been identified by experts of the field correlating risk values with respect to visual inspection of the sequences. At this point the conflict is recorded with the corresponding GPS position. Finally the tuple composed of GPS position, RI-level and class of conflict are stored on a database. These data collected automatically can be successively studied by experts to take actions on the road infrastructure.

#### 4.7. Implementation considerations

In this paper a number of hardware and software components were combined in order to achieve the overall output of the system: the identification of places, in an urban environment, where safety is compromised in some way, due to the detection of many Traffic Conflicts in those very places.

The design idea behind all the presented framework, was that each part could be replaced with another one that produce the same output in the pipeline.

The novelty of the proposed framework is the knowledge about how to combine and customize simple components in order to obtain the final result. Thus we had to make some choice for each of the presented hardware and software: the hardware components were chosen in order to have the best quality for input data; while the software modules were chosen starting from open-source projects and easiness-to-customization. The open-source feature not only gave us short time-to-deploy but also guarantees the

**Table 2**

Confusion matrix related to the three possible Traffic Conflict classes.

		Predicted		
		Intersection	Pedestrian	Car following
Actual	Intersection	100%	0%	0%
	Pedestrian	12.5%	87.5%	0%
	Car following	0%	0%	100%

reproducibility of the overall proposed framework. To this end we selected some basic computer vision algorithms like HOG for pedestrian detection and mean-shift as a code-base for tracking.

HOG was selected for pedestrian detection being fast and already implemented into the OpenCV C++ Library. Moreover it performed well with the help of depth-data pre-processing described in 4.5 used to eliminate false positives.

The purpose of the proposed framework is not to present a good pedestrian technique (between many described in a recent survey [31] HOG is still a good technique) but a Pedestrian Crossing detection algorithm that combined with all other modules achieves the objective.

As regards the mean-shift, we needed a fast and lightweight tracking algorithm with an easy implementation to be used as a code-base for customization. Our customization introduces a lot of “a-priori” knowledge from environment in order to achieve best results.

Other algorithmic solutions could be easily replaced by end-user however, taking advantage from the insights learned from this paper, will be able to build a system to achieve same or better results than those presented in next section.

## 5. Experimental results

The validation of the classifiers described in Section 4.5 has been carried out on a test set, labelled by an expert and obtained from 1 h of acquisition containing 108 elements as candidate TCs. The results of classification on the test set are reported in the confusion matrices in Tables 2 and 3. The accuracy of the cascade of classifiers stated at an accuracy of 99.97% for TCs classification with respect to the three classes and an accuracy of 99.07% for the TC prediction.

The system described in this paper has been setup on city buses of the public transportation company operating in the city of Catania, Italy. The line identified for our tests runs through the main streets of the city centre. Data used for the experiments have been acquired on four week days for a total of 16 h, 4 drivers and different weather and light conditions. 528 TCs were found. A summary of the results obtained with the proposed framework is shown in Table 4. It is possible to note that although most

**Table 3**

Confusion matrix related to Traffic Conflict Prediction.

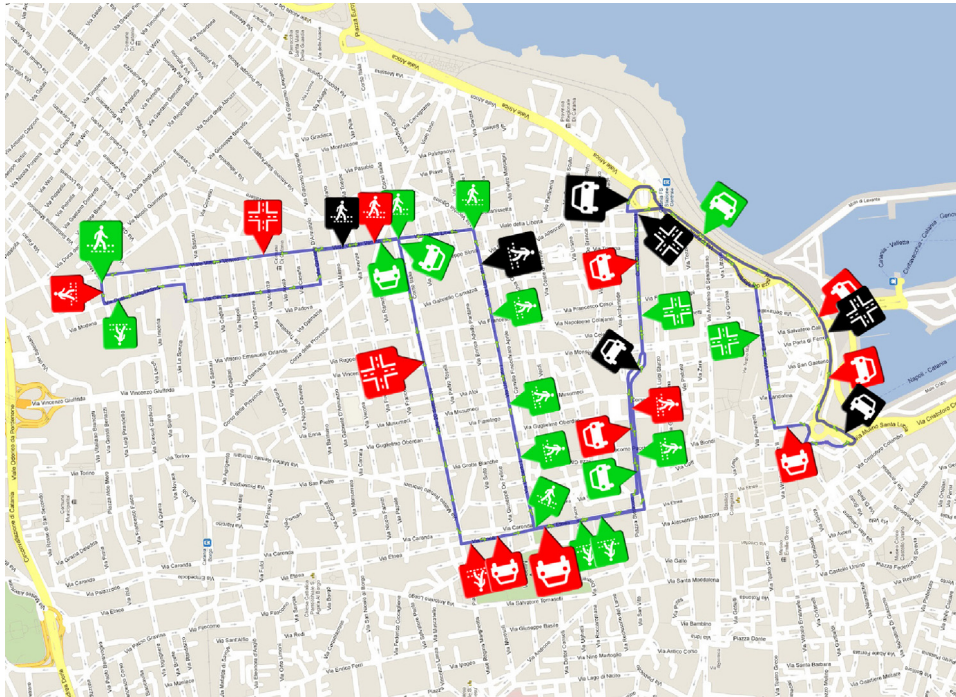
		Predicted	
		TCs	Not a TC
Actual	TCs	100%	0%
	Not a TC	1.3%	98.6%

**Table 4**

Distribution (%) of collected TC classes detected per RI-level.

TC class	Class	Low RI	Medium RI	High RI
Pedestrian	46.34	63.15	21.05	15.79
Car following	36.58	26.67	53.33	20
Intersection	17.07	28.57	28.57	42.85





**Fig. 10.** The visual output of the presented framework is a graphical representation of all TC occurrences with different class, frequency and RI-level.

recognised conflicts are related to pedestrian, the most frequent high RI-level occurs in the case of intersection conflict.

The conflicts recognised by our framework, are graphically reported on a map where each conflict is shown as a pin with different color and size as well as with a symbol that represents the TC class:

- Green means low RI;
- Red means medium RI;
- Black means high RI.

The size of the marker is proportional to the number conflicts of the same TC class and RI level detected in that place.

The final map obtained with our framework is shown in Fig. 10. The augmented map gives an idea of the distribution of the risk for road safety in the analysed urban area and allows to immediately detect locations at which to intervene. Moreover by using our framework it is possible to analyse the results of an intervention and have a feedback of the actual safety improvement.

To further demonstrate the goodness of our framework we compared the obtained results with the crash database of the Public Transport Company AMT of Catania: the places in which there is an high RI (black in Fig. 10) are locations where each month there is at least one crash recorded confirming the low safety level of the places.

## 6. Conclusion and future works

Given the unpredictability of driver behaviours and the random and rare nature of crashes, it is becoming increasingly apparent that the necessary data collection should be obtained from naturalistic approaches like the Traffic Conflict Technique. The framework described in this paper is based on the Traffic Conflict Analysis theory in conjunction with Computer Vision algorithms. It can be used as a decision support system for analysts and governments to improve road safety and intervene to prevent accidents instead of waiting for them to happen and collect statistical data to make a decision. A possible future work can

involve a behavioural analysis of the driver based on wearable technologies [32]. Also vision systems acquiring information on drivers through video analysis can give further information to the system to better understand what are the causes of a conflict and then to learn how to better prevent an accident. Finally all of the data collected with these kind of systems can be used to make a driving assistive technology that infer how dangerous is a certain road and alert the driver itself or change vehicle parameters in order to prevent certain predefined condition that can create a conflict or an accident.

## Acknowledgements

The authors wish to thank the Italian Ministry of Economic Development for the financial support of this research within the program 'Industria 2015', the Public Transport Company AMT Catania for the kind availability of the city bus and the research team of the Department of Environmental and Civil Engineering of University of Catania led by professor Cafiso.

## References

- [1] E. Hauer, *Observational Before/After Studies in Road Safety. Estimating the Effect of Highway and Traffic Engineering Measures on Road Safety*, (1997).
- [2] T.A. Dingus, S. Klauer, V. Neale, A. Petersen, S. Lee, J. Sudweeks, M. Perez, J. Hankey, D. Ramsey, S. Gupta, et al., *The 100-Car Naturalistic Driving Study, Phase II—Results of the 100-Car Field Experiment*, (2006) Tech. Rep..
- [3] S. Kaplan, M.A. Guvensan, A.G. Yavuz, Y. Karalurt, *Driver behavior analysis for safe driving: a survey*, *IEEE Trans. Intell. Transp. Syst.* 16 (6) (2015) 3017–3032.
- [4] A. Williamson, T. Chamberlain, *Review of On-road Driver Fatigue Monitoring Devices*, University of New South Wales, 2005, pp. 43–52.
- [5] J. Autey, T. Sayed, M.H. Zaki, *Safety evaluation of right-turn smart channels using automated traffic conflict analysis*, *Accid. Anal. Prev.* 45 (2012) 120–130.
- [6] A. Pascale, M. Nicoli, F. Defflorio, B. Dalla Chiara, U. Spagnolini, *Wireless sensor networks for traffic management and road safety*, *Intell. Transp. Syst. IET* 6 (1) (2012 March) 67–77.
- [7] P. St-Aubin, N. Saunier, L. Miranda-Moreno, *Large-scale automated proactive road safety analysis using video data*, *Transp. Res. Part C: Emerg. Technol.* 58 (Part B) (2015) 363–379 *big Data in Transportation and Traffic Engineering*.
- [8] S. Jackson, L. Miranda-Moreno, P. St-Aubin, N. Saunier, *Flexible, mobile video camera system and open source video analysis software for road safety and behavioral analysis*, *Transp. Res. Rec.: J. Transp. Res. Board* 2365 (2013) 90–98.

- [9] C.M. Farmer, B.B. Kirley, A.T. McCartt, Effects of in-vehicle monitoring on the driving behavior of teenagers, *J. Saf. Res.* 41 (1) (2010) 39–45.
- [10] J.S. Hickman, R.J. Hanowski, Use of a video monitoring approach to reduce at-risk driving behaviors in commercial vehicle operations, *Transp. Res. Part F: Traffic Psychol. Behav.* 14 (3) (2011) 189–198.
- [11] W.J. Horrey, M.F. Lesch, M.J. Dainoff, M.M. Robertson, Y.I. Noy, On-board safety monitoring systems for driving: review, knowledge gaps, and framework, *J. Saf. Res.* 43 (1) (2012) 49–58.
- [12] E.H. Gazali, Monitoring Erratic Driving Behaviour Caused by Vehicle Overtaking Using Off-the-Shelf Technologies, (2010) .
- [13] M. Fazeen, B. Gozick, R. Dantu, M. Bhukhiya, M.C. González, Safe driving using mobile phones, *IEEE Trans. Intell. Transp. Syst.* 13 (3) (2012) 1462–1468.
- [14] S. Chigurupati, S. Polavarapu, Y. Kancharla, A.K. Nikhath, Integrated computing system for measuring driver safety index, *Int. J. Emerg. Technol. Adv. Eng* 2 (6) (2012).
- [15] P. Singh, N. Juneja, S. Kapoor, Using mobile phone sensors to detect driving behavior, *Proceedings of the 3rd ACM Symposium on Computing for Development*, ACM, 2013, pp. 53.
- [16] S. Battiato, S. Cafiso, A. Di Graziano, G. Farinella, O. Giudice, Road traffic conflict analysis from geo-referenced stereo sequences, *Image Analysis and Processing – ICIAP 2013, Lecture Notes in Computer Science*, vol. 8156, Springer Berlin Heidelberg, 2013, pp. 381–390.
- [17] J.A. Misener, C. Nowakowski, X.Y. Lu, T. Koo, J.D. Margulici, J. Spring, Onboard Monitoring and Reporting for Commercial Motor Vehicle Safety (Report No. FMCSA-RRT-07-030), Federal Motor Carrier Safety Administration, Washington, DC, 2007.
- [18] H.W. Heinrich, *Industrial Accident Prevention*, McGraw-Hill, New York, NY, 1932.
- [19] C. Hyden, The Development of Method for Traffic Safety Evaluation: The Swedish Traffic Conflict Technique, Lund Institute of Technology, Sweden, 1987 Tech. Rep. Bulletin 70.
- [20] S. Cafiso, A. Garcia, R. Cavarra, M.A.R. Rojas, Pedestrian crossing safety improvements: before and after study using traffic conflict techniques, 4th International Symposium on Highway Geometric Design (2010) 2–5.
- [21] D. Gettman, L. Pu, T. Sayed, S.G. Shelby, Surrogate Safety Assessment Model and Validation, (2008) Tech. Rep..
- [22] J. Woodfill, G. Gordon, R. Buck, TYZX DeepSea high speed stereo vision system, *Conference on Computer Vision and Pattern Recognition Workshop, CVPRW '04*, June 2004 (2004) 41.
- [23] R. Zabih, J. Woodfill, Non-parametric local transforms for computing visual correspondence, in: J.-O. Eklundh (Ed.), *Computer Vision – ECCV '94, Lecture Notes in Computer Science*, vol. 801, Springer Berlin Heidelberg, 1994, pp. 151–158.
- [24] Video Codec Specification, (2013) . (accessed 24.01.18) <http://www1.mplayerhq.hu/michael/ffv1.html>.
- [25] D.L. Mills, Internet time synchronization: the network time protocol, *IEEE Trans. Commun.* 39 (10) (1991) 1482–1493.
- [26] S. Cafiso, A. Di Graziano, O. Giudice, G. Pappalardo, Using GPS data to detect critical events in motorcycle rider behaviour, *Int. J. Mobile Netw. Des. Innov.* 5 (4) (2014) 195–204.
- [27] DLIB C++ Library, (2018) . (accessed 24.01.18) <https://github.com/davisking/dlib>.
- [28] OpenCV C++ Library, (2018) . (accessed 24.01.18) <https://github.com/opencv/opencv>.
- [29] D. Comaniciu, V. Ramesh, P. Meer, Kernel-based object tracking, *IEEE Trans. Pattern Anal. Mach. Intell.* 25 (May (5)) (2003) 564–577.
- [30] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2005. CVPR 2005, vol. 1, June (2005) 886–893.
- [31] P. Dollar, C. Wojek, B. Schiele, P. Perona, Pedestrian detection: an evaluation of the state of the art, *IEEE Trans. Pattern Anal. Mach. Intell.* 34 (4) (2012) 743–761.
- [32] A. Furnari, G.M. Farinella, S. Battiato, Recognizing personal contexts from egocentric images, *Workshop on Assistive Computer Vision and Robotics (ACVR) in Conjunction With ICCV* (2015).