PUVAME - New French Approach for Vulnerable Road Users Safety

Olivier Aycard, Anne Spalanzani, Manuel Yguel, Julien Burlet, Nicolas Du Lac, Arnaud de la Fortelle, Thierry Fraichard, Hicham Ghorayeb, Mikael Kais, Christian Laugier, et al.

PUVAME - New French Approach for Vulnerable Road Users Safety

Olivier Aycard*, Anne Spalanzani*, Manuel Yguel*, Julien Burlet*, Nicolas Du Lac§,
Arnaud De La Fortelle‡, Thierry Fraichard*, Hicham Ghorayeb†, Michael Kais‡,
Christian Laugier*, Claude Laurgeau†, Gilles Michel§, David Raulo*, Bruno Steux†
*GRAVIR-IMAG & INRIA RA - Grenoble - FRANCE
Email: FirstName.LastName@imag.fr
†Ecole des Mines de Paris - Paris - FRANCE
Email: FirstName.LastName@ensmp.fr
‡INRIA Rocquencourt - Rocquencourt - FRANCE
Email: FirstName.LastName@inria.fr
§INTEMPORA - Paris - FRANCE
Email: FirstName.LastName@intempora.com

***Abstract*— In France, about 33% of roads victims are VRU[1]. In its 3^{rd} framework, the french PREDIT[2] includes VRU Safety. The PUVAME project was created to generate solutions to avoid collisions between VRU and Bus in urban traffic. An important part of these collisions take place at intersection or bus stop. In this paper, we detail the hardware and software architecture designed and developped in the project. This solution is based on offboard cameras observing particular places (intersection and bus stop in our case) to detect and track VRU present in the environment. The position of the bus is also computed and a risk of collision between each VRU and the bus is determined. In case of high risk of collision, the bus driver is warned. The HMI to warn the bus driver is also described. Finally, some experimental results are presented.**

I. Introduction

In France, about 33% of roads victims are VRU. In its 3^{rd} framework, the french PREDIT includes VRU Safety. The PUVAME project was created to generate solutions to avoid collisions between VRU and Bus in urban traffic. This objective will be achieved by:

- Improvement of driver's perception capabilities close to his vehicle; This objective will be achieved using a combination of offboard cameras, observing intersections or bus stops, to detect and track VRU present at intersection or bus stop, as well as onboard sensors for localisation of the bus;
- Detection and assessment of dangerous situations, analyzing position of the vehicule and of the VRU and estimating their future trajectories;
- Triggering alarms and related processes inside the vehicule;
- Integration on experimental vehicules.

The project started on october 2003 and will end in april 2006.

The partners are:

- INRIA[3].
- Ecole des Mines de Paris - Centre de Robotique
- Connex-Eurolum
- Robosoft
- ProBayes
- Intempora
- INRETS LESCOT

In this paper, we present the PUVAME project. In next section, we detail the accident analysis done by Connex-Eurolum in 2003 and also describe the chosen use cases. Section III presents the experimental platform used to evaluate the solutions we propose. Section IV details the architecture of the system. Experimental results are reported in section V. We give some conclusions and perspectives in section VI.

II. Accident analysis+scenario intersection

In the scope of the project, we've analysed accidents occured in 2003 between vulnerables (pedestrians, cyclists) and buses in a french town. 3 kinds of accidents arised from this study. In 25.6% of the accidents, the vulnerable was struck while the bus was leaving or approaching a bus stop. In 38.5% of the accidents, the pedestrian was attempting to cross at an intersection when he was struck by a bus turning left of right. Finally, 33.3% of the accidents occured when the pedestrian lost balance on the sidewalk when he was running for the bus, or was struck by a lateral part of a bus or one of its a rear view mirrors. It was also noticed that in all these cases, most of the impacts occured on the right side or the front of the bus.

In the aim of reducing these kinds of accidents, we proposed 3 scenarios to reproduce the most frequent accidents' situations and find ways to overcome the lack of security. The first scenario aims at reproducing vulnerables struck while the bus arrives or leaves its bus stop (see Figure 1(a)). The second scenario aims at reproducing vulnerables struck at an intersection (see Figure 1(b)). The third scenario aims at reproducing vulnerables struck by the lateral part of the bus, surprised by the sweeping zone. In this article, we proposed

[1] Vulnerable Road Users
[2] Programme de Recherche et d'Innovation dans les Transports Terrestres
[3] French National Institute for Research in Computer Science and Control

to focus on the first two scenari when a pedestrian is struck by the right part of the bus. In these 2 cases, it has been decided to use fixed cameras placed at the bus stop or at the road junction in order to detect potentialy dangerous behaviors. As most of the right part of the bus is unseen by the driver, it is very important to give him information about the fact that a vulnerable is placed in this blind spot. The cameras will cover the entire blind zone.

(a) (b)

Fig. 1. (a) The bus arrives or leaves its bus stop. The vulnerables situated in the blind spot near the bus are in danger because they are not seen by the driver and have a good probabiliy to enter in collision with the bus; (b) The pedestrian crosses at an intersection when the bus turns right.

Information given by cameras will be analysed and merged with information about the position of the bus. A collision risk estimation will be done and an interface will alert the driver about the danger. A more detailled description of the process will be done in section IV. Next section presents the experimental site set up at INRIA Rhône-Alpes where these two scenari are under test.

III. ParkNav platform

The experimental setup used to evaluate the PUVAME system is composed of 2 distinctive parts: the ParkView platform used to simulate an intersection or a bus stop and the cycab vehicule used to simulate a bus.

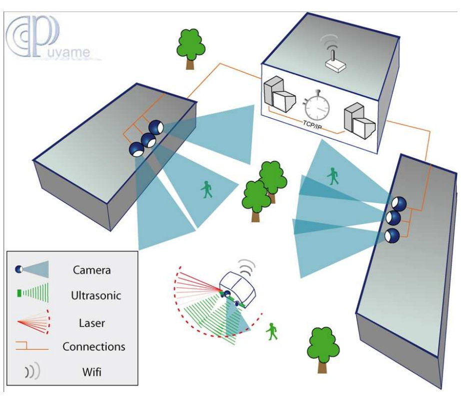
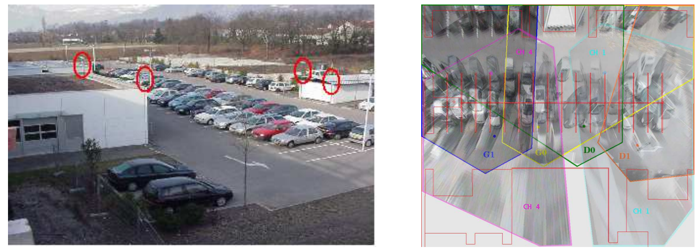


Fig. 2. The ParkNav system overview

The RTMaps software from Intempora S.A.[4] has been used both in the cycab vehicle and the infrastructure (see figure 2) in order to timestamp, synchronize, record and process the data on such a distributed system. This way, on-board sensors as well as infrastructure cameras were timestamped according to a single shared clock, which enabled efficient development of data-fusion algorithms.

The ParkView platform is composed of a set of six off-board analog cameras, installed in a car-park setup such as their field-of-view partially overlap (see figure 3), and three Linux(tm) workstations in charge of the data processing, connected by a standard Local Area Network.



(a) (b)

Fig. 3. (a) Location of the cameras on the parking; (b) Field-of-view of the cameras projected on the ground.

The workstations are running a specifically developped client-server software composed of three main parts, called the *map server*, the *map clients* and the *connectors* (figure 4).

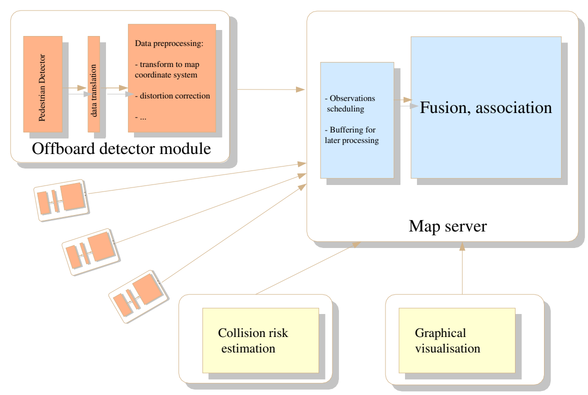


Fig. 4. The ParkView platform software organization

The map server: processes all the incoming observations, provided by the different clients, in order to maintain a global high-level representation of the environment; this is where the data fusion occur. A single instance of the server is running.

The connectors: receive the raw sensor-data, perform the pre-processing, and send the resulting *observations* to the map server. Each of the computer connected with one or several sensors is running such a *connector*. For the application described here, all data preprocessing basically consist in detecting pedestrians. Therefore, the video stream of each camera is processed independantly by a dedicated detector. The role of the detectors is to convert each incoming video frame to a set of bounding rectangles, one by target detected in the image plane (Fig 8(a)). The set of rectangles detected at

[4] http://www.intempora.com

a given time constitutes the detector observation, and is sent to the map server.

Since the fusion system operates in a fixed coordinate system, distinct from each of the camera's local systems, a coordinate transformation must be performed. For this purpose, each of the cameras has been calibrated beforehand. The result of this calibration consists in a set of parameters:

- the intrinsic parameters contain the information about the camera optics and CCD sensor: the focal length and focal axis, the distorsion parameters,
- the extrinsic parameters consist of the homography matrix: this is the 3x3 homogenous matrix which transform the coordinates of an image point to the ground coordinate system.

The map clients: connect to the server and provide the users with a graphical representation of the environment; they can also process this data further and perform application-dependant tasks. For example, in a driving assistance application, the vehicle on-board computer will be running such a client specialized in estimating the collision risk. In this paper, the graphical client provides the user with the global occupancy grid overlayed on the map of the car park (Fig 9).

A. The CyCab vehicule

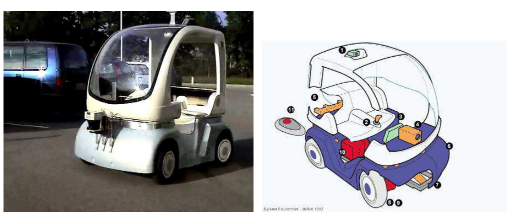


Fig. 5. The CyCab vehicule

The CyCab (figure 5) has been designed to transport up to two persons in downtown areas, pedestrian malls, large industrial or amusement parks and airports, at a maximum of 30km/h speed. It has a length of 1.9 meter, a width of 1.2 meter and weights about 300 kg. It is equipped with 4 steer and drive wheels powered by four 1 kW electric motors. To control the cycab, we can manually drive it with a joystick or fully automatic operate it. It is connected to the ParkView platform by a wireless connection: we can send it motors commands and collect odometry and sensors data: the Cycab is considered as a client of the ParkNav platform. It perceives the environment with a sick laser used to detect and avoid the obstacles.

IV. Architecture of the System

In this section, we detail the PUVAME software architecture (figure 6) we choose for the intersection and bus stop scenario. This architecture is composed of 4 main parts:

1) First of all, the images of the different offboard camera are used to estimate the position and speed of each pedestrian present in the crossroadmap;

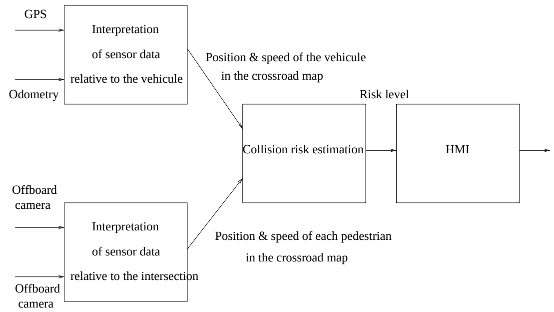


Fig. 6. Architecture system

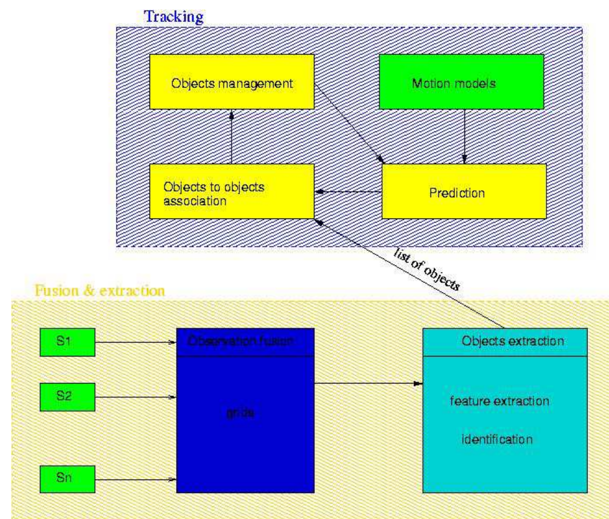


Fig. 7. Architecture of the pedestrians tracker

2) The odometry and the Global Positionning System are used to determine the position and speed of the vehicule in the crossroad map;
3) Third, the position and speed of the different objects present in the intersection are used to estimate and a risk of collision between the bus and each pedestrian;
4) Finally, the level of risk and the direction of the risk are sent to the HMI inside the bus to warn the bus driver.

In the next subsections, we detail these 4 modules.

A. Interpretation of sensor data relative to the intersection

Our objective is to have a robust perception using multi-sensor approaches to track the different pedestrians present in the car park. The whole architecture is depicted in figure 7. This architecture is composed of two distinctive parts: a Fusion and Extraction level and a Tracking level. In the first level, we perform fusion of detected pedestrians given by the different cameras to build a map of the current environment (ie, a snapshoot of the current environment). In a second step, using this map, we search the pedestrians currently present in the environment. Finally, in the tracking level, we associate this list of pedestrians with the list of pedestrians previously present in the environment. In this section, we describe the different modules of the architecture.

B. Fusion and extraction level

1) Pedestrian detector: To detect VRUs present at the intersection, a pedestrian detector subsystem is used. The detector is composed of three components: the first component consists in a foreground segmentation based on Multiple Gaussian Model as in [7]. The second component is a sliding window binary classifier for pedestrians using AdaBoost-based learning methods [1], [8]. The third component is a tracking algorithm using image based criteria of similarity.

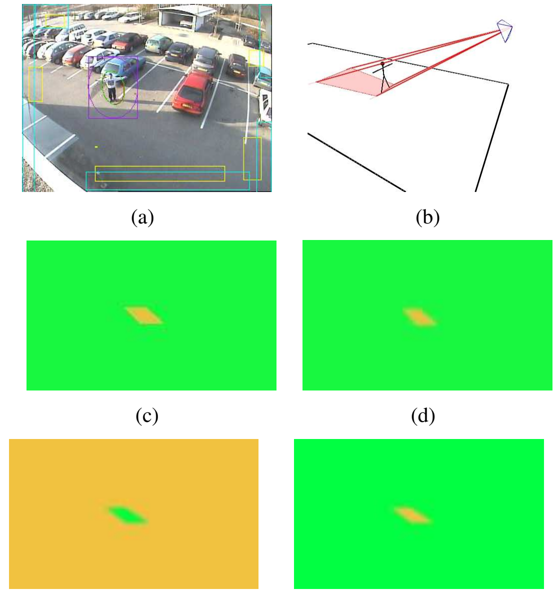
2) Occupancy grid: Occupancy grids is a generic framework for multi-sensor fusion and modelling of the environment. It has been introduced by Elfes and Moravec [3] at the end of the 1980s. An occupancy grid is a stochastic tesselated representation of spatial information that maintains probabilistic estimates of the occupancy state of each cell in a lattice. The main advantage of this approach is the ability to integrate several sensors in the same framework taking the inherent uncertainty of each sensor reading into account, in opposite to the Geometric Paradigm whose method is to categorize the world features into a set of geometric primitives [2]. The alternative that OGs offer is a regular sampling of the space occupancy, that is a very generic system of space representation when no knowledge about the shapes of the environment is available. On the contrary of a feature based environment model, the only requirement for an OG building is a bayesian sensor model for each cell of the grid and each sensor. This sensor model is the description of the probabilistic relation that links sensor measurement to space state, that OG necessitates to make the sensor integration.

The construction of the sensor model associated with the detector observations given by different cameras is detailed in [9]. In this paragraph, we only give an overview of the construction of this sensor model.

The problem is that detector observations give information in the image space and that we search to have knowledge in the ground plan. We solve this problem projecting the bounding box in the ground plan, supposing that the ground is a plan, all the VRU stand on the ground and the complete VRUs is visible for the camera.

Also, we first search to segment the ground plan in three types of region: occupied, occulted and free zones using the bounding boxes informations. Then we introduce an uncertainty management, using a gaussian convolution, to deal with the position errors in the detector. Finally, we convert this information into probability distributions. Figure 8 illustrates the whole construction of the sensor model.

a) Results: Figure 9 shows the same pedestrian seen by two cameras. The red area corresponds to the most probable position of the pedestrian: this area is the result of the fusion of the two yellow areas given the two cameras. The 3 green areas around the pedestrian correspond to the fusion between the occluded area of one camera with the free area of the other one. The area seen as free by the two cameras has a very low probability of occupancy. The 4 areas seen as free by one camera and out of the field of view of the second camera have a low probability of occupancy.



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

Fig. 8. (a) An image of a moving object acquired by one of the offboard video cameras and the associated bounding boxe found by the detector. (b) The occulted zone as the intersection of the viewing cone associated with the bounding boxe and the ground plan. (c) The associated ground image produce by the system. (d) Ground image after gaussian convolution with a support size of 7 pixels. (e) Probability of the ground image pixel value, knowing that the pixel corresponds to an empty cell: $P(Z|\text{emp})$ for each cell. (f) Probability of the ground image pixel value, knowing that the pixel corresponds to an occupied cell: $P(Z|\text{occ})$ for each cell.

3) Object extraction: Once an occupancy grid is obtained, we want to extract the possible objects (VRUs) which are likely located in regions with high occupation probability Object-regions may have arbitrary shapes and are generally discriminant from background. From these characteristics, we apply a threshold segmentation method.

First, an adaptive threshold is computed based on a discrete histogram of cell occupation probability values and the threshold is chosen as the mean value of the histogram. We use this threshold to transform the grid into a binary image where positive pixels represent occupied areas. In the next step a two pass segmentation algorithm is applied to extract all 4-connected groups of cells. Each connected group corresponding to a possible object is finally approximated by an ellipse represented by mean value and covariance matrix of the corresponding region (see Figure 10).

C. Tracking part

1) Prediction: Each VRU present in the environment is tracked using a Kalman filter [5]. The state vector is represented by both position and velocity of the VRU and the predicted state is computed using a constant velocity dynamical model.

2) Object to Object association: To update the position of each VRU using Kalman filter, we first need to associate the observations extracted from the occupancy grid to the

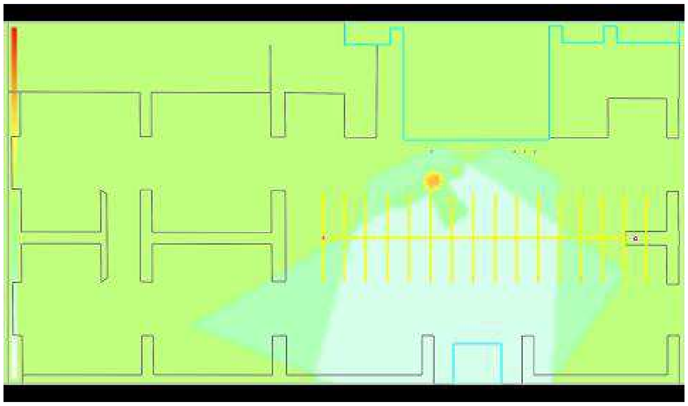


Fig. 9. The resulting probability that the cells are occupied after the inference process with two cameras.

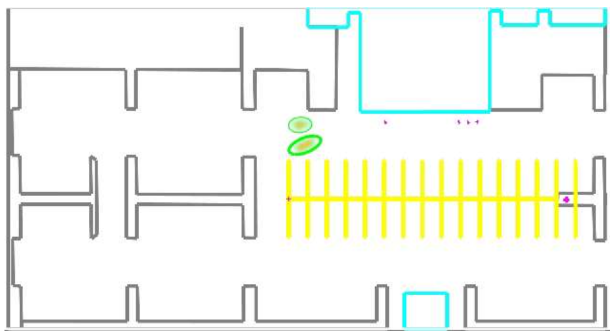


Fig. 10. detection of objects approximated with ellipses

predicted positions. As there could be at most one observation associated to each given VRU: a gating procedure is first applied to reduce number of possible assignments, then a global nearest neighbor data association method is used [6]. The association is also useful to manage the list of VRUs present in the environment as described in the next paragraph.

3) Object management: Each VRU is tagged with a specific ID, its position in the environment and the associated velocity. At the beginning of the process, the list of VRU present in the environment is empty. The result of the association phase is used to update this list. Several cases could appear:

1) An observation is associated to a VRU: the position and velocity of this VRU is estimated with a Kalman filter, the predicted state and this observation;
2) A VRU has no observation associated to itself: the reestimated position and velocity of this VRU are given by the predicted state;
3) An observation is not associated to any VRU: a new temporary VRU ID is created, its position is initialized at the value of the observation and its velocity is set to 0. To avoid to create VRU corresponding to false alarms, the temporary VRU is only confirmed (ie, becomes a definitive VRU) if it is seen during 3 consecutive instants.

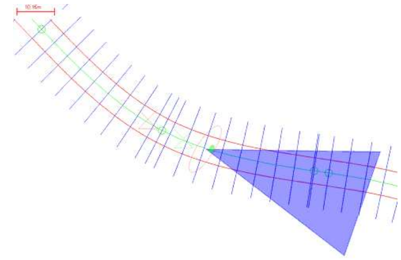


Fig. 11. Agregation of road information from a GIS and position of the vehicle

As we are using off-board cameras observing always the same environment, 2 conditions are needed to delete a VRU of the list: it has to be unseen (ie, no observation has been associated to it) for at least the last 3 instants and its estimated position should be outside the intersection.

D. Interpretation of sensor data relative to the vehicule

The goal of the localization module [4] is to compute the position and the orientation of the bus at the intersection with respect to a fixed reference frame. Our approach relies on fusion of exteroceptive information with proprioceptive information. Exteroceptive measurements provide position and orientation of the Cycab with respect to the reference frame, based on the Global Positionning System (GPS). Since Direct line of view with four satellites is not guaranteed in the cities due to buildings GPS data is hybridized with proprioceptive data. A gyrometer was used to measure the angular velocity along the vertical axis of the Cycab while the longitudinal speed was measured using encoders.

Fusion of exteroceptive and proprioceptive information was based on an Extended Kalman Filter (EKF). The prediction step computes the estimation of the pose of the vehicle by integrating the longitudinal speed and the angular speed using a bicycle model. In this case, obviously the imprecision of the estimation of the pose increases. When a GPS data is received, Correction step is resumed.

Sensors like GPS are usually affected by some latency i.e. some time between the instant of measurement and the instant the data is delivered to the controller in charge of computing the position estimation. To handle this problem, two actions were carried out. First each data is precisely tagged in Coordinated Universal Time (UTC) citeKais06, since GPS position is also tagged in UTC, all data are in the same temporal frame. The second action consisted in designing an EKF with some memory functionality. With our approach measurements (GPS positions), commands (speeds) and intermediate states are stored in memory for a while. With a simple EKF, the data is fused in the filter when it is available. With memory EKF, the data is fused using the time of the measurement.

This figure represents the aggregation of the road model coming from a Geographic Information System and the vehicle localization information. Road boundaries are represented by the red curves while median axis is represented by the green

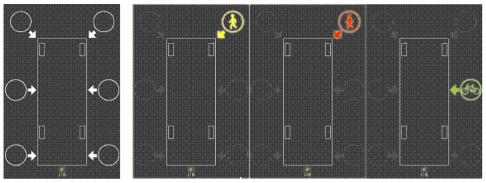


Fig. 12. interface (left) and differents warning functions

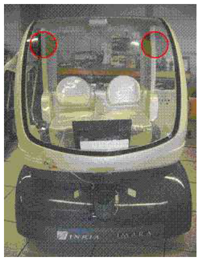


Fig. 13. the cycab and the position of the 2 speakers

curve. The estimated positions for a 99% probability using a bivariate normal distribution from two localization systems are highlighted with the red ellipse and the green filled ellipse (inside the red allipse). Last received GPS positions are highlighted by the two crosses.

E. Collision Risk Estimation

The risk of collision is computed with the Closest Point of Approach and the Time to Closest Point of Approach method. The Closest Point of Approach is the point where two moving objects are at a minimum distance. To compute this distance, we suppose that these two moving objects are moving at a constant distance. So at each time, we are able to compute the distance between these two objects. To know when this distance is minimal, we search the instant where the derivative of the square of this distance is null. This instant is named the Time to Closest Point of Approach and the respective distance is named the Closest Point of Approach.

In our application, at each time, we compute the Time to Closest Point of Approach and the respective distance (ie, Closest Point of Approach) between each VRU and the bus. If this distance is below a given threshold in less than 5 seconds, we compute the orientation of the VRU relative to the bus and generate an alarm for the HMI.

F. Warning interface

The interface (figure 12) is made of 2 parts. One is the bus outline (center of the interface), and the other is the possible position of vulnerables (the 6 circles). These circles can be fullfil by warning pictograms. These pictograms show the place where a vulnerable could be struck by the bus : in front, middle, back, left or right side of the bus. The representation need to be very simple in order to be understood immediately by the driver. Moreover, 2 speakers are added in order to warn rapidly the driver about the danger (figure 13).

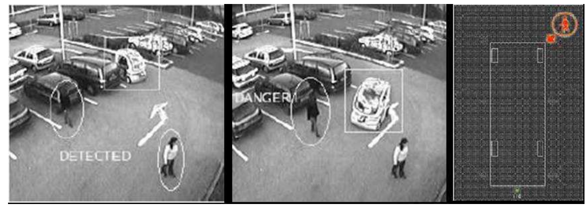


Fig. 14. (left) 2 pedestrians are crossing the road. They are detected by our system. The cycab is arriving turning right. (center) The system estimates that there is a high probability of collision between the pedestrian starting crossing the road. (right) It alerts the driver with a signal on the interface

V. Experimental results

We reproduced the scenario shown (see Figure 1(b)) in our experimental site. The bus is a cycab, 2 pedestrians crossing the road are detected by our system. One is in danger because it has a high probability to be struck by the cycab. The driver is alerted by the interface that a pedestrian is potentially in danger (see Figure 14).

VI. Conclusion

The proposed system to reduce accidents between VRU and buses is based on challenging technologies. The software architecture and the modules composing this architecture have been detailed in this article. All these modules have been developped and tested. A preliminary release of the whole system has been used to present some experimental results. The consortium is now working on the integration of the final modules on the parknav platform.

Acknowledgment

PUVAME is a project of the “Security oriented technologies” operational group of the PREDIT. The partners of PUVAME thank the French PREDIT for supporting the work of this project. The partners of PUVAME thank the French CNRS ROBEA ParkNav project for partially supporting the work of this project.

References

[1] Y. Abramson, B. Steux, and H. Ghorayeb. Yef real-time object detection. 2005.

[2] H. Cramer, U. Scheunert, and G. Wanielik. Multi sensor fusion for object detection using generalized feature models. In *International Conference on Information Fusion*, 2003.

[3] Alberto Elfes. Using occupancy grids for mobile robot perception and navigation. *Computer*, 22:46–57, June 1989. ISSN:0018-9162.

[4] M. Kais, D. Millescamps, D. Betaille, B. Lusetti, and A. Chapelon. A multi-sensor acquisition architecture and real-time reference for sensor and fusion methods benchmarking. In *IEEE International Conference on Intelligent Vehicules*, June 2006.

[5] E. Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME-Journal of Basic Engineering*, 82:35–45, 1960.

[6] S.Blackman and R. Popoli. *Design and Analysis of Modern Tracking Systems.* A K Peters, 1999.

[7] Chris Stauffer and W. Eric L. Grimson. Learning patterns of activity using real-time tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):747–757, 2000.

[8] P. Viola and M. Jones. Fast and robust classification using asymmetric adaboost and a detector cascade. 2002.

[9] M. Yguel, O. Aycard, D. Raulo, and C. Laugier. Grid based fusion of offboard cameras. In *IEEE International Conference on Intelligent Vehicules*, 2006.