Frontal Object Perception using radar and mono-vision

R. Omar Chavez-Garcia*, Trung-Dung Vu*, Julien Burlet†, Olivier Aycard*
*University of Grenoble1 - FRANCE
Email: FirstName.LastName@imag.fr
†TRW Conekt, Birmingham - ENGLAND
Email: FirstName.LastName@trw.com

***Abstract*—In this paper, we detail a complete software architecture of a key task that an intelligent vehicle has to deal with: frontal object perception. This task is solved by processing raw data of a radar and a mono-camera to detect and track moving objects. Data sets obtained from highways, country roads and urban areas were used to test the proposed method. Several experiments were conducted to show that the proposed method obtains a better environment representation, i.e., reduces the false alarms and miss detections from individual sensor evidence.**

I. Introduction

Perceiving or understanding the environment surrounding a vehicle is a very important step in driving assistance systems or autonomous vehicles. Recently, there have been considerable research efforts focusing on specific aspect of this problem [1], [2] in the frame of several european projects[1].

However, for many aspects of perception, there still remains many open questions. In the frame of the european project Interactive, the goal is to

In this paper, we detail the software architecture of one module of this architecture: the frontal object perception is dedicated to detect and track the moving objects located in front of the ego-vehicle. This module uses the raw data of a radar and a camera as inputs.

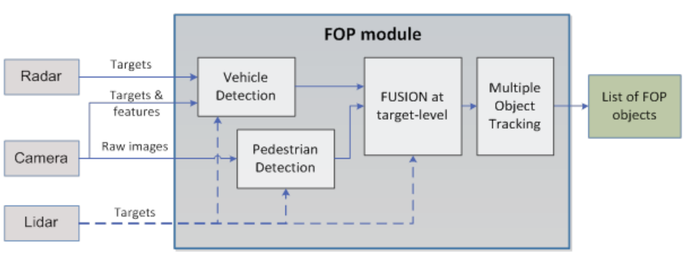

Fig. 1. General architecture of the Frontal Object Perception.

Figure 1 shows the general architecture of the frontal object perception module for one of the ego-vehicle involved in the interactIVe project. This architecture is composed of three levels of processing that are described in the next sections:

1) first of all, raw data of radar and mono-vision are processed to detect moving objects. Actually, these processings are done in two steps: first of all, each sensor delivers informations about moving objects that it has detected. In a second step, and secondly these informations are used to detect moving pedestrians and moving vehicles.
2) the Fusion processing which takes as an input the list of the detected objects (pedestrians and vehicles) provided by both kind of sensors and delivers a fused list of detected objects.
In most of the works related to fusion [3] [4], the fusion is done after tracking. The main advantage of this strategy is that the fusion could be designed independently of the sensors. This is very convenient to design generic methods to perform sensor data fusion. On the other hand, high level fusion has several drawbacks: to be able to perform fusion, we have to wait until an object is tracked which can take several frames. Another important problem is when an object is detected, it is sometimes not created for tracking due to numerous false alarms or mis-detections. To overcome these difficulties, we will perform the fusion at the detection level (ie, before tracking). The basic idea is to perform fusion using the list of detected objects provided by each sensor to decrease the level of false alarms or mis-detections and hence to improve the quality of tracking.
3) the Tracking processing module which takes as an input the fused list of objects and delivers a list of tracked objects.

The rest of the paper is organized as follows. In the next section, we present the demonstrator used for this work and sensors installed on it. In section III, we describe the sensor processing done with radar and mono-vision. In sections IV and V, we detail our work on fusion and tracking. Experimental results are reported in section VI. We conclude this work in section VII.

II. Vehicle demonstrator

A car demonstrator part of the interactIVe European project is used in order to obtain data sets from different situations. The process of data acquisition focus in three scenarios: highway, countryside, and urban areas. The TRW Conekt demonstrator car is a Fiat Stilo previously used in the PReVENT-SASPENCE project. It is equipped with a sensor

[1]http://www.intersafe-2.eu, www.prevent-ip.org
[1]http://www.interactive-ip.eu

array composed of the TRW AC100 medium range radar mounted below the registration plate and the TRW TCam camera, positioned below the rear view mirror, providing lane detection and raw image for video processing. Also vehicle ego motion is filtered and provided through the CAN bus. Figure 2 shows images of the interactIVe demonstrator used to perform the experiments.

Fig. 2. Images from the TRW Conekt demonstrator.

The radar sensor is medium range radar with a detection range up to 150m, a field of view of $\pm 8°$, and an angular accuracy of $0.5°$. The camera has on-board processing and image recognition routines embedded, its frame rate is 30Hz.

III. Radar and Monovision processing

This section details the processings that are done on the radar and on the mono camera: the low level radar processing delivers a list of detected moving objects and the mono camera processing delivers information about the detected object located in front of the ego vehicle.

In a second step, two post processing modules are included in order to detect specific objects of interest: vehicles and pedestrians. These modules, based on radar and visual information, intent to reduce the number of false alarms from radar sensor and the pedestrian miss detections. The input data for these modules is composed of a list of targets detected by the radar and camera sensors. The output of both modules are a list of detected objects (pedestrians or vehicles).

In the next subsections, we give details about these 4 modules.

A. Radar processing

The mid-range TRW radar contains an internal processing able to detect static and moving objects having a radar cross-section similar to a car. The list of these objects, called targets, is passed through CAN bus. Radar data is given as a list of n targets detected in the radar field of view. Each element of the list include the range, azimuth and relative speed of the detected target. As the sensor will produce a return for each object with a significant radar cross section, targets may correspond to static objects or objects different from vehicle, producing false positives. In a similar way, 'weak objects' like pedestrian can not always be detected consequently producing miss detections. Therefore, it is necessary to address the issues in the next stages of the processing.

B. Mono camera processing

The TRW camera also contains an internal processing which computes an headway detection and send through CAN bus, call VideoCAN for clarification purposes. This camera-embedded process outputs the position of the detected object in front of the ego-vehicle giving the centre point of the object in term of range and lateral relative position, and an estimation of its width. While this data contains very few false detections, we have to deal with the approximated accuracy of the detection and some miss detection is complex cases.

C. Vehicle detector

This block of the architecture aims to use both radar and video outputs at two different steps to perform robust vehicle detection. Radar sensors have a good range resolution and a crude azimuth estimation and video sensors are able to give a precise lateral estimation while having an uncertain range estimation. A two-stage fusion process gains the advantages of both sensors while suppressing their drawbacks.

Radar output is used for gating purposes to confirm the detections at each stage and assign a good range and velocity to the detected object. The camera output is then used to confirm the initial detection and refine the object lateral position.

Furthermore, to improve the efficiency of the detector, detections are integrated through time using a multi-object tracking approach. Each track state estimation and prediction is performed using a Kalman Filter. This simple filter allows a fast computation while producing sufficient results for this detection stage for which the aim is only to produce robust detection and not perform very precise tracking.

Figure 3 shows the approach taken and the different steps used in the multi-object detection and tracking with sensor fusion. Blue coloured blocks represent different sensor inputs, red blocks the different outputs and grey represents all the functional blocks.

The first step of the detection cycle is to use the previous track list and the radar output to perform association. For this association stage, a nearest neighbour approach is used and allows tagging of each track as "not confirmed", "associated" or "un-associated radar object" (new track).

In the next step, the camera is used to compute a histogram of edges for each track. Using this histogram, the most probable object boundary is computed, allowing the objects lateral position to be obtained. Furthermore, according to the histogram shape and density, a likelihood function computes the probability that the current track is a vehicle. The use of this likelihood function has three advantages at this stage of the method:

- The first is that is permits unassociated tracks to be kept with the radar, allowing us to cope with radar misdetections.

Fig. 3. Pedestrian Detector work flow. Inputs are depicted in blue and outputs in red.

- The second is that it allows assignment and refinement at each cycle of the probability that the track is a vehicle.
- The last is that it prevents spurious measurement from the radar creating a new track which does not match a vehicle histogram pattern.

In a later step, tracks which have been confirmed are updated according to the associated radar track and the video histogram search result. The new track state estimation is performed using the Kalman filter associated for each track. After this state, the list of tracks is outputted as the final detection set and used in the next steps of the architecture. Moreover, Tracks that have not been confirmed by histogram search have their deletion counter incremented. If this deletion counter reaches a defined level, the track is marked for deletion and will be deleted at the end of the cycle. If the deletion counter has not been reached, the track is still considered valid and is treated as an associated track for the rest of the process.

Finally, in a last stage, the prediction of the Kalman filter is run for each valid track. The Kalman prediction is made according to the track velocity (given by the radar), range and lateral position and the ego-vehicle motion. This prediction will be used for the gating with the radar in the next cycle.

By following this approach, robust vehicle detection is performed by using the complementarities of the radar and the video sensors. Figure 4 (a) shows an output example from the vehicle detector. It can be seen that this post processing module still has some miss detections and false alarms that have to be corrected in order to provide a better list of detected objects.

D. Pedestrian detector

Pedestrian detection is a difficult task in computer vision domain, there exist several works regarding this problem. A good visual feature selection for pedestrian representation is needed in order to achieve a good detection rate, avoiding false alarms and reducing miss detections.

The pedestrian detector module scans the image using a sliding window of fixed size to detect the pedestrians. A modified version of histogram of oriented gradients (HOG) features powers the pedestrian representation at training and detection time. Several images, extracted from the Daimler Pedestrian Benchmark data sets [5], were used as training datasets for the training process.

The classification process is made by a boosting-based classifier. Boosting is a powerful learning concept, which provide a solution to the supervised classification learning task. It combines the performance of many $weak$ classifiers (slightly correlated with the true classification) to produce a powerful $group$ [6]. Weaks classifiers are only required to perform better than chance, hence they can be very simple and computationally inexpensive.

The boosted model is based on N training examples (x_i, y_i) with $x_i \in R^k$ and $y_i \in \{-1, +1\}$. x_i is a K-component vector. Each component encodes a feature relevant for the learning task. The desired two-class output is encoded as -1 and $+1$.

We use the standard two-class discrete AdaBoost algorithm reviwed in [7]. Each training sample is initially assigned the same weight. Next a weak classifier $f_m(x)$ is trained on the weighted training data. Its weighted training error and scaling factor c_m is computed. The weights are increased for training samples, which have been misclassified. All weights are then normalized, and the process of finding the next weak classifier continues for another $M - 1$ times. The final classifier $F(x)$ is the sign of the weighted sum over the individual weak classifiers. Decision trees are usedas weak classifiers in this boosting scheme.

The list of targets given by this detector includes a bounding box surrounding each detected pedestrian. A camera-ground transformation process is performed, using the horizon line provided by the camera, to convert the image reference system to the evidential grid reference system. Figure 4 (b) shows an output example obtained from the pedestrian detector.

Fig. 4. Outputs from the post processing modules: (a) green bounding boxes represent the detected vehicles; (b) yellow bounding boxes represent the pedestrians detected by the pedestrian detector.

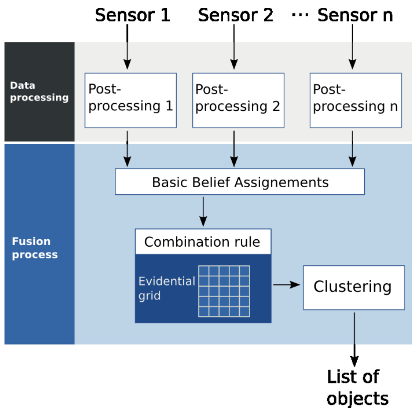

Fig. 5. General scheme of the fusion approach.

IV. Fusion process

DS occupancy grids have been used since several years ago [8], [9], [10], [11], these grids are an alternative to the widely used Bayesian occupancy grids proposed in [12]. Both kind of grids are a discrete representation of the real world, they aim to quantify the occupancy or emptiness of each cell using information from sensors. Usually, in a Bayesian grid, a cell has two possible states: empty (E) or occupied (O). However, in DS theory there are two more states: *ignorance* and *conflict* states. Evidence is obtained from the observations that support one or a set of possible states. The set $\Omega = \{E, O\}$ is called the *frame of discernment* and the number of states for each cell is the power set of this frame: 2^{Ω}.

DS theory uses basic belief assignments (BBA) to initialize each state in 2^{Ω}. A BBA maps the evidence for each subset in 2^{Ω} to a real value, this value means the support of the evidence for a particular state. A BBA function is described as follows:

$$\begin{aligned} & m(\emptyset) = 0, \\ & \textstyle\sum_{A \subseteq \Omega} m(A) = 1. \end{aligned} \tag{1}$$

In order to combine different sources of evidence a combination rule is required. Several fusion operators have been proposed in the evidential framework concerning scenarios with different requirements. One of the widely used is that proposed by Dempster [13]. Dempster's rule of combination assumes independence and reliability of both sources of evidence.

Figure 5 shows the proposed method that performs the fusion at detection level. This receives, as inputs, the list of detected objects from several sensor processing modules. Iteratively, the fusion process i) takes one of the inputs and represented it into a temporary evidential grid using a sensor model to do so; 2) subsequently, it fuses the current evidential grid with the temporary one; until all the inputs are processed. A clustering process is needed after the fusion process to identify the objects from the final evidential grid. All the evidence this method uses comes from the current state of the sensors and neither previous nor predicted information is included. Several sensors and post-processing modules can be attached to extend the proposed architecture.

Following the Dempster's rules definition from [13], we obtain equation 2 for two mass functions m_1 and m_2, where m_1 represents the current state of the environment and m_2 the new available data from one of the sensors. The fusion process assumes independence of the cells and combines the evidence cell by cell from the two evidential grids.

$$\begin{array}{ll} m_1 \oplus m_2(\{F\}) = & m_1(\{F\})m_2(\{F\}) + \\ & m_1(\{F\})m_2(\{F,O\}) + \\ & m_1(\{F,O\})m_2(\{F\}) \\ \hline m_1 \oplus m_2(\{O\}) = & m_1(\{O\})m_2(\{O\}) + \\ & m_1(\{O\})m_2(\{F,O\}) + \\ & m_1(\{F,O\})m_2(\{O\}) \\ \hline m_1 \oplus m_2(\{F,O\}) = & m_1(\{F,O\})m_2(\{F,O\}) \\ \hline K_{1 \oplus 2} = & m_1(\{F\})m_2(\{O\}) + \\ & m_1(\{O\})m_2(\{F\}) \end{array} \tag{2}$$

we can see how evidence is supporting each possible state from Ω, for example state $\{F\}$, is supported by beliefs from $m_1(\{F\})$, $m_2(\{F\})$, $m_1(\{F,O\})$, $m_2(\{F,O\})$ while ignorance state $\{F,O\}$ is just supported by evidence from $m_1(\{F,O\})$ and $m_2(\{F,O\})$. Reliability assumption from both sources allows us to use the belief from $\{F,O\}$ state to support other states in Ω. Each one of the mass functions obtained from the combination rule is normalized using the conflict factor $K_{1 \oplus 2}$.

A. Sensor models

The proposed fusion process requires these BBAs as inputs, several sensor models are proposed to generate the corresponding BBAs and populate the grids.

1) Radar: Radar sensor givens a list of n targets detected in the radar field of view. We assume that all the area outside of the field of view of the radar is $unknown$, likewise the triangular area behind the targets, this means that we have no evidence to support this area is occupied or empty. All the area inside the target area is considered $occupied$. All the remaining area is considered as $free$. A mapping process was implemented to transform real target coordinates into a common frame of reference.

Figure 6 a) shows an example of the evidential grid for radar sensor. Here we can see all the cases explained above. Even if graphically the cells inside the target area are marked as $occupied$, their vector of mass functions have real values.

The evidence we obtain from the sensor data is defined by a BBA and applied to each cell depending on its occupancy state. The BBA that the proposed method uses is defined as follows.

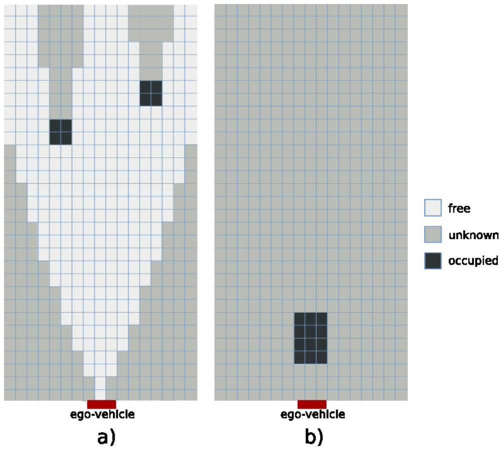

Fig. 6. Evidential grid for a) radar sensor; and b) camera sensor after applying its sensor model.

$$
\begin{array}{ll}
m(\{F\}) = 1 - \lambda_{fa}, & \\
m(\{O\}) = 0, & \\
m(\{F,O\}) = \lambda_{fa}, & if\ cell\ is\ free \\
m(\{\emptyset\}) = 0 & \\
\hline
m(\{F\}) = 0, & \\
m(\{O\}) = 1 - \lambda_{md} \times \gamma, & \\
m(\{F,O\}) = \lambda_{md} \times \gamma, & if\ cell\ is\ occupied \\
m(\{\emptyset\}) = 0 & \\
\hline
m(\{F\}) = 0, & \\
m(\{O\}) = 0, & \\
m(\{F,O\}) = 1, & if\ cell\ is\ unknown \\
m(\{\emptyset\}) = 0 &
\end{array}
\tag{3}
$$

where λ_{fa} and λ_{md} are confidence factors in $[0,1]$ that represent the false alarms and miss detections ratio for this sensor, doing this we are representing the uncertainty from the sensor observations.

2) Camera: The camera sensor, from now on known as videoCAN, obtains the detected object (only one) in front of the ego-vehicle. We perform the same transformation process as we did for radar sensor. Here we consider two possible areas: the area inside the detected object is considered as evidence for the occupied state; the area outside the detected object is considered as $unknown$ because there is no evidence about the state of each cell and because of the sensor just detects the object in the very front regardless other possible existing objects.

Figure 6 b) represents the evidential grid for videoCAN sensor data. Using the assumptions from the videoCAN sensor evidence we define the BBA as will be stated next.

$$
\begin{array}{ll}
m(\{F\}) = 1 - \lambda_{fa}, & \\
m(\{O\}) = 0, & \\
m(\{F,O\}) = \lambda_{fa}, & if\ cell\ is\ free \\
m(\{\emptyset\}) = 0 & \\
\hline
m(\{F\}) = 0, & \\
m(\{O\}) = 1 - \lambda_{md}, & \\
m(\{F,O\}) = \lambda_{md}, & if\ cell\ is\ occupied \\
m(\{\emptyset\}) = 0 & \\
\hline
m(\{F\}) = 0, & \\
m(\{O\}) = 0, & \\
m(\{F,O\}) = 1, & if\ cell\ is\ unknown \\
m(\{\emptyset\}) = 0 &
\end{array}
\tag{4}
$$

here we use two factors to represent the uncertainty of the measures for *occupied* (λ_{fa}) or $free(\lambda_{md})$ states.

3) Post processing modules: A similar sensor model to the VideoCAN is proposed to map the information from the vehicle detector module, but instead of taking just one object as occupied area, it takes the area inside each detected vehicle. The rest area is taken as *empty* because we suppose the vehicle detector detects all the objects of interest. Two individual factors indicate the false alarms and miss detections for this post processing module.

In order to use the output list provided by the pedestrian detector module we need to transform this list into evidence in the evidential grid. For this reason we use a similar sensor model to that used for the vehicle detector module: based in the area occupied by the detected objects and in the area where no objects are detected.

B. Clustering process

A clustering process is used to find the objects within the final grid. The main parameter used to perform this process is the highest mass value of each cell. Figure 7 shows the result of the clustering process illustrating three objects founded after the proposed fusion method. A cell is considered to be part of an object if its highest mass value comes from O set. All the cells with highest O value inside a neighborhood are part of the same object. All the cells with highest $\{O,F\}$ value surrounding an object are part of the same object. Cells with highest F value are not part of any object. The area composed with only cells with highest value O is called the area of maximum belief. The area surrounding an object and composed only for cells with highest O,F value is called the uncertainty area.

V. TRACKING PROCESS

In general, the multi objects tracking problem is complex: it includes the definition of tracking methods, but also association methods and maintenance of the list of objects currently present in the environment. Bayesian filters are usually used to solve tracking problem. These filters require the definition of a specific motion model of tracked objects to predict their positions in the environment. Using the prediction and observation update combination, position estimation for

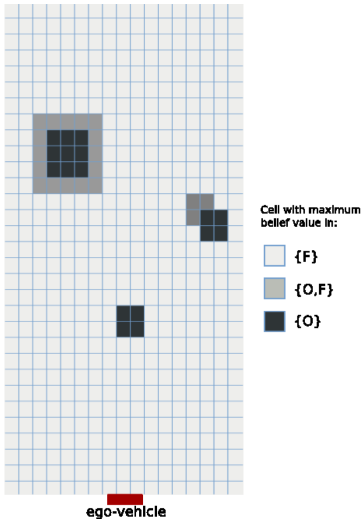

Fig. 7. Clustering process result. Three objects of different size are founded.

each object is computed. In the following we explain the components of our tracking module.

A. Data Association

This step consists of assigning new objects of fused list to the existing tracks. Since in the current work we are concerned with many objects moving in different directions: they may be crossing or wating to cross in a direction perpendicular to the oncoming vehicles, for example a vehicle waiting to turn left etc. We have used MHT [14] approach to solve the data association problem. To further control the growth of tracks trees we need to use some pruning technique. We have chosen the N-Scans pruning technique to keep the track trees to a limit of N.

B. Track Management

In this step tracks are confirmed, deleted or created using the m-best hypotheses resulting from the data association step. New tracks are created if a new track creation hypothesis appears in the m-best hypothesis. A newly created track is confirmed if it is updated by objects detected in current frames after a variable number of algorithm steps (one step if the object was detected by both laser and stereo vision otherwise in three steps). This implies that the spurious measurements which can be detected as objects in the first step of our method are never confirmed. To deal with non-detection cases, if a non-detection hypothesis appear (which can appear for instance when an object is occluded by an other one) tracks having no new associated objects are updated according to their last associated objects and for them next filtering stage becomes a simple prediction. In this way a track is deleted if it is not updated by a detected object for a given number of steps.

C. Filtering

Since there may be different types of objects (vehicles, motor bikes, pedestrains etc) moving in different directions using different motion modes, a single motion model based filtering technique is not sufficient. To address the tracking problem, we have used an on-line adapting version of Interacting Multiple Models (IMM) filtering technique. The details of this technique can be found in our other published work [15]. We have seen that four motion models (constant velocity, constant acceleration, left turn and right turn) are sufficient to successfully track objects. We use four Kalman filters to handle these motion models. Finally the most probable trajectories are copmuted by taking the most probable branche and we select one unique hypothesis for one track tree.

D. Tracking Output

The output of tracking process consists of position and velocity information of ego vehicle alongwith a list of tracks. A track is a moving object with its position, orientation and velocity information as well as a reference to its instance in the previous frame.

VI. Experimental Results

Several real datasets were taken as input for the proposed method. These datasets include highway, rural, and urban scenarios. The objectives of the experiments are to show if using evidential fusion at detection level, from several sensors we can obtain a better representation of the environment, this means a reduction of false alarms and miss detections.

A. Experiments setup

Regarding the scheme showed by figure 5 the experimental architecture is composed of two sensors: camera and radar. Four post-processing modules use the information from the sensors in order to supply four evidence sources to the fusion method. Two of these modules are embedded in the sensor and the other two are the vehicle and sensor detectors.

The size of the evidential grid is defined by the grid resolution, i.e. the number of cells per meter. For the experiments we use a grid resolution of 8 cells per meter. Uncertainty values for the fusion method were set experimentally.

B. Experimental results

Figures 8, 9, 10 show the results obtained by the proposed fusion method after the clustering process is applied. A car is represented by a green rectangle; its position uncertainty is shown as a red circle; black blocks represent the area which higher mass value is set in the occupy state; light gray areas means higher values for ignorance states. Small black squares without bounding shapes stand for pedestrians. Individual evidential grids are shown as well in the bottom of the image, from left to right: videoCAN, radar, vehicle detector and pedestrian detector; the last one is the fusion result before the clustering process. The displayed data in figure 8 represents a highway scenario.

Most of the detected vehicles are obtained from the vehicle detector and radar sensor. Both evidences support the occupancy of these particular cells. VideoCAN just support one of the detected objects and add evidence to the unknown state

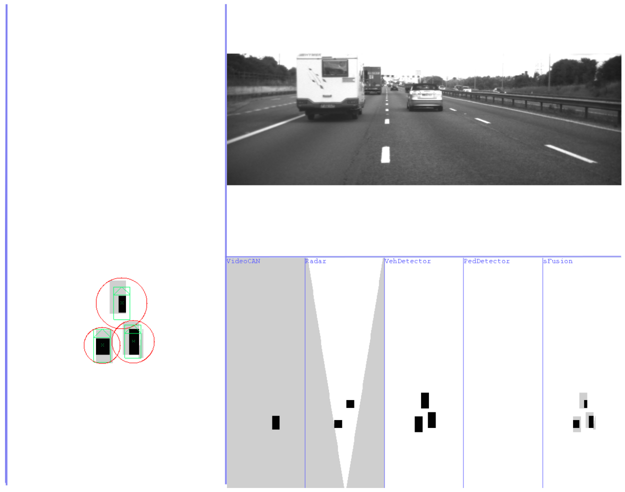

Fig. 8. Fusion results from a highway scenario (left). Individual evidential grids from the different sensors (bottom-right). Image of the current scenario (top-right).

for the rest of the cells. We can see in this example that all the objects of interest are detected.

In country-mile areas the information from sensors becomes more noisy, radar targets are commonly false alarms. Using evidence from the other sensors the fusion method can reduce the number of false alarms. Figure 9 evinces how false alarms are reduced after applying the proposed fusion method.

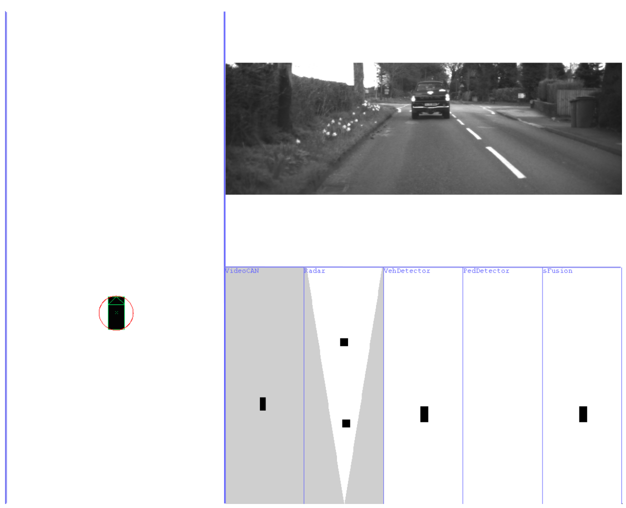

Fig. 9. Fusion results from a country road scenario (left). Individual evidential grids from the different sensors (bottom-right). Image of the current scenario (top-right).

An example over an urban area is shown in the figure 10. Here one can see how several vehicles are detected and how pedestrians interact in the scenario. Not all the vehicles in the scenario are detected because of the lack or conflict in the evidence provided by the sensors. For an ADAS the objects of interest surrounding the system are a priority in order to reason and take some actions, we can see that most of the near vehicles, surrounding the demonstrator, are detected thanks to the complementary evidence.

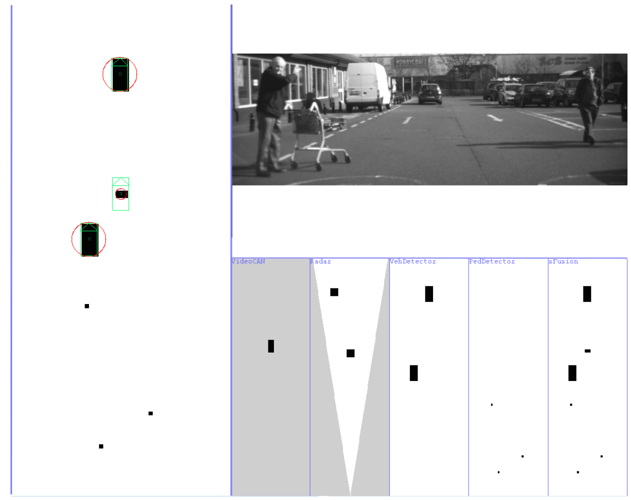

Fig. 10. Fusion results from an urban scenario (left). Individual evidential grids from the different sensors (bottom-right). Image of the current scenario (top-right).

Just one sensor provides information regarding pedestrian, for this reason this information is not taken into account into the fusion process hence it is passed directly to the output.

Within the interactiVE project the perception platform is considered to be real time if its processing time does not exceed 200ms. Average results show, empirically, that the proposed FOP approach suits the real time requirement usually attached to ADAS applications.

Results can be improved using information from beam-based sensors that could allow to have more particular evidence from points not covered, resumed from high-level data or help in conflict situations. Laser sensors might provide interesting evidence as input for the proposed method improving the detection of vehicles and pedestrians.

Some videos showing our results could be found at *http://membres-liglab.imag.fr/aycard/html/Demos/iv2012_fop.html*

VII. Conclusions

In this paper, we detailed our software architecture for the Frontal Object Perception of the european project Interactive. This module uses radar and mono-vision to detect and track moving objects. Moreover, a detection level fusion between the two sensors, based on Dempster Shafer Theory, is done to improve the detection.

Several experiments were conducted regarding three particular scenarios: highway, country side and urban areas. In all the scenarios, most of the objects of interest were detected and false alarms from individual post-processing modules were reduced. Experiments shows as well that the proposed method performs the Frontal Object Perception task in real time.

The two next steps are: the use of a laserscanner (see figure 1) to detect and track moving objects [15] and fuse with radar and mono-vision, and the add of classfication information about moving objects like pedestrians, cars, bicycles, bus and trucks.

ACKNOWLEDGMENT

This work was also supported by the European Commission under interactIVe, a large scale integrating project part of the FP7-ICT for Safety and Energy Efficiency in Mobility. The authors would like to thank all partners within interactIVe for their cooperation and valuable contribution.

REFERENCES

[1] S. Pietzsch, T. Vu, J. Burlet, O. Aycard, T. Hackbarth, N. Appenrodt, J. Dickmann, and B. Radig, "Results of a precrash application based on laser scanner and short range radars," *IEEE Transactions on Intelligent Transport Systems*, vol. 10, no. 4, pp. 584–593, 2009.

[2] C. Shooter and J. Reeve, "Intersafe-2 architecture and specification," in *IEEE International Conference on Intelligent Computer Communication and Processing*, 2009.

[3] H. Baltzakis, A. Argyros, and P. Trahanias, "Fusion of laser and visual data for robot motion planning and collision avoidance," *Mach. Vision Appl.*, vol. 15, no. 2, pp. 92–100, 2003.

[4] S. Pagnottelli, S. Taraglioy, P. Valigi, and A. Zanelay, "Visual and laser sensory data fusion for outdoor robot localisation and navigation," in *12th International Conference on advanced robotics (ICAR)*, pp. 171–177, 2005.

[5] M. Enzweiler and D. M. Gavrila, "Monocular pedestrian detection: Survey and experiments," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, pp. 2179–2195, December 2009.

[6] T. Hastie, R. Tibshirani, and J. Friedman, *The elements of statistical learning: data mining, inference, and prediction*, vol. 27. Springer, 2009.

[7] J. Friedman, T. Hastie, and R. Tibshirani, "Additive logistic regression: a statistical view of boosting," *Annals of Statistics*, vol. 28, p. 2000, 1998.

[8] D. Pagac, E. Nebot, and H. Durrant-Whyte, "An evidential approach to map-building for autonomous vehicles," *IEEE Transactions on Robotics and Automation*, vol. 14, no. 4, pp. 623–629, 1998.

[9] T. Yang and V. Aitken, "Evidential Mapping for Mobile Robots with Range Sensors," *Test*, 2005.

[10] W. Huadong, M. Siegel, R. Stiefelhagen, and J. Yang, "Sensor fusion using dempster-shafer theory [for context-aware hci]," in *Instrumentation and Measurement Technology Conference, 2002. IMTC/2002. Proceedings of the 19th IEEE.*, vol. 1, pp. 7 – 12, 2002.

[11] M. Skutek, D. T. Linzmeier, N. Appenrodt, and G. Wanielik, "A precrash system based on sensor data fusion of laser scanner and short range radars," *2005 7th International Conference on Information Fusion*, p. 8 pp., 2005.

[12] A. Elfes, "Using occupancy grids for mobile robot perception and navigation," *Computer*, vol. 22, pp. 46–57, June 1989.

[13] P. Smets, "The transferable belief model for belief representation," *In Gabbay and Smets*, vol. 6156, no. Drums Ii, pp. 1–24, 1999.

[14] D. B. Reid, "An algorithm for tracking multiple targets," *IEEE Transactions on Automatic Control*, vol. 24, no. 6, 1979.

[15] T.-D. Vu and O. Aycard, "Lased-based detection and tracking moving object using data-driven markov chain monte carlo," in *IEEE International Conference on Robotics and Automation (ICRA)*, (Kobe, Japan), May 2009.