

# Radar Signature in Multiple Target Tracking System for Driver Assistant Application

Haisheng Liu

School of Electronics and Information  
Nantong University, Jiangshu, China  
email: haisheng.liu@ntu.edu.cn

Smail Niar

LAMIH, Université de Valenciennes  
le Mont-Houy, Valenciennes, France  
email: smail.niar@univ-valenciennes.fr

**Abstract**—This paper presents a new Driver Assistant System (DAS) using radar signatures. The new system is able in one hand to track multiple obstacles and on the other hand to identify obstacles during vehicle movements. The combination of these two functions on the same DAS gives the benefits of avoiding false alarms. Also, it makes possible to generate alarms that take into account the identification of the obstacles. The obstacle tracking process is simplified thanks to the identification stage. Hence, our low cost FPGA-based System-on-Chip is able to detect, recognize and track a large number of obstacles in a relatively reduced time period. Our experimental result proves that a speed up of 32% can be obtained compared to the standard system.

**Index Terms**—FPGA, Driver Assistance System, Radar signature, MTT, System-on-Chip

## I. INTRODUCTION

Driver Assistance Systems (DASs) are an increasingly important class of automotive applications in nowadays commercial vehicles. They improve greatly road safety in stressful driving conditions such as at night or in bad weather. Adaptive Cruise Control [1], radar aided automatic proximity control and navigation systems are examples of well-known range of high-tech DAS applications. In the field of collision avoidance system, the MTT (Multiple Target Tracking) system is considered to be an efficient solution that provides good obstacle detection and tracking capacities.

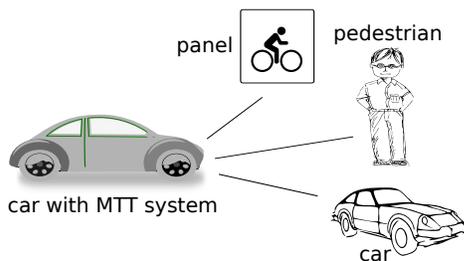


Fig. 1. Obstacle detection system based on radar sensor

The detection functionality is generally realized by one or more sensors (radar or camera) around the host vehicle (Fig. 1). In this paper, we only consider the radar detection due to its good detection performance especially while driving with poor visibility. Thanks to the embedded MTT application, the driver can be alerted in real-time by a 3D audio alarm in case of danger.

Historically, Application Specific Integrated Circuit (ASIC) occupied the first place in embedded systems for the automotive industry because of their cost-effective silicon solution [2]. Increasing levels of complexity and computational demands in the automotive applications forced a move to more powerful yet cost-effective processors [3]. As a promising alternative to ASIC solution, reconfigurable platforms have been adopted to implement complex automotive subsystems [4].

Indeed, existing driver assistance systems do not support complex applications including fragile road users (such as pedestrians or cyclists) or under rapid changing of weather and lighting conditions, because of their limited functionalities or cost constraints for large-scale automotive utilization. The reconfigurable hybrid SW-HW solution based on FPGA technology, for Field-Programmable Gate Arrays, emerges as one of the most promising platforms for DAS application. The frequently changed functions are implemented on programmable processors whereas the most time-consuming functions are mapped on hardware accelerators using reconfigurable circuits.

Two types of DAS have been proposed during the last years: the recognition system and the MTT system. The recognition system categorizes the obstacles by using the inherent radar signatures. This functionality is called ORS for *Obstacle Radar Signature*. The ORS is then used to generate an adequate alarm related to the type of the obstacle. The MTT function is crucial for driver assistance applications such as collision avoidance, intelligent cruise control and automatic parking. However, the computational requirements of a MTT application increases significantly as the number of targets becomes important. Our contribution in this paper is to design a full DAS where the ORS and the MTT work in a coordinated manner. Indeed, this cooperation is very effective to reduce the system cost and to enhance the system accuracy.

This paper is organized as follows. Section II presents the related research works. Section III explains the radar signatures and their utilization in Multiple Target Tracking system. The baseline MTT architecture is given in Section IV. Section V exposes the gate and signature checks. The time profile results are presented in Section VI.

## II. RELATED WORK

In recent years, various types of DAS have been proposed. Among the most popular DAS functionalities, we can cite:

Adaptive Cruise Control [1], Lane Keep Assistance System [5], Parking Assistance System [6], Obstacle Detection and Avoidance System [7]. Most of the existing systems have either limited functionalities or are too costly for a large-scale automotive utilization. These systems are implemented by different hardware and/or software architectures. From the hardware point of view, these systems range from dedicated hardwired ASIC to pure programmable processors.

To offer a good performance/flexibility/cost trade-off, designers have proposed either to use multi-processor system-on-chips (MPSoC) or hardwired FPGA-based circuit [4]. Imap-CAR [8] and EyeQ2 [9] systems are two examples of full programmable processors that are dedicated to automotive security applications using vision system. Both of the architectures provide support for a specific set of real-time data intensive applications. Therefore, these systems are unable either to accommodate new applications or to adapt the hardware to different scenarios. In addition, the AutoVision processor [10] is a dynamically reconfigurable MPSoC prototype for video-specific pixel processing.

Also using the radar device, the research work in [11] is the closest work of ours. The proposed MPSoC architecture demonstrated the feasibility of using software-programmable processor cores, up to 20 soft-cores, to execute a complex application and it still meets the real-time constraints. However, it also demonstrated the high cost, in area and resource utilization, associated with a software-only implementation. This is particularly evident with the Kalman Filtering (KF) block where dedicated processors are used to execute the Kalman filtering code for each target. Even when multiple targets could be tracked by a single Kalman filtering block, the cost of implementing an entire processor in the logic fabric of an FPGA to only support a single, dedicated function may be too high.

A more cost-effective solution would be to use a hybrid system that combines dedicated filtering hardware blocks with software-programmable processor cores. The authors in [12] proposed the utilization of DPR (Dynamic Partial Reconfigurable) technology to implement the Kalman filtering function. Hence, the multiple soft-cores for data filtering are replaced by the reconfigurable hardware logics and the different filters can be charged and uncharged according to the driving context. Compared to the above works, to the best of our knowledge, our work is the first that uses the radar signature to optimize the MTT architecture and to reduce the execution time.

### III. RADAR SIGNATURE FOR OBSTACLE IDENTIFICATION

In this section, we first give a brief review of the Multiple Target Tracking (MTT) system. Secondly, the obstacle signature is discussed as well as its implementation.

#### A. Radar Signature and MTT application

In Fig. 1, three different obstacles: a pedestrian, a panel and a car, are detected in front of the host vehicle. The conventional MTT system does not make any distinction among these obstacles if only considering the obstacle presence. The

audio alarm is generated in the same manner regardless of what the obstacle is. However, it is highly desired that the DAS can yield different alarms according to the degree of dangerousness. That is why it is necessary to identify the obstacle to achieve that performance. On the other hand, the identification helps to eliminate the false alarms for those obstacles considered not dangerous.

Fig. 2 shows graphically the block diagram of a radar-based MTT system with obstacle signature enhancement. Here, the detection in our experimentation is performed by a commercial AC20 TRW radar whose scan period is set to 25 ms. In each scan, the MTT application is executed to deal with the measurement data in real-time. After the sampling phase and the analog to digital conversion at the reception, the detection block outputs the obstacle position consisting of the distance and the angle, denoted by  $(d, \theta)$ . The signature id ( $S$ ) is obtained thanks to the identification stage. The MTT computation is then based on the position  $(d, \theta)$  and the respective signature  $S$ . Lastly, the corresponding audio alarm is to be reconstituted according to the MTT computation results, which as stated takes into account both the obstacle position and the respective signature.

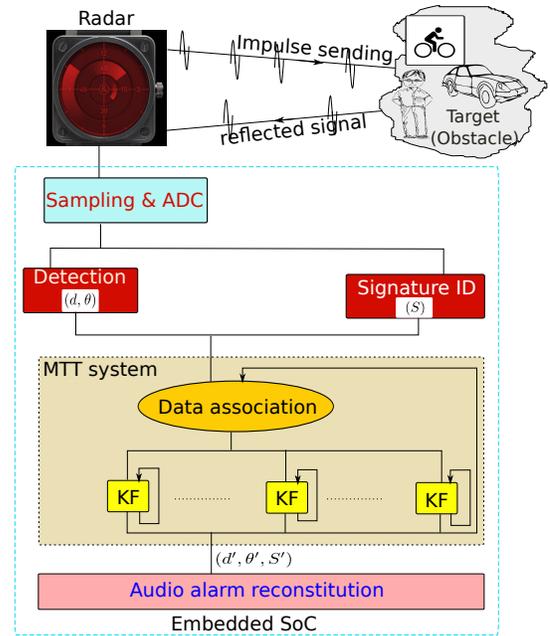


Fig. 2. Block diagram of a radar-based MTT system with signatures

#### B. Radar signature and its implementation

It is demonstrated in [13] that the impulse response from a target can be expressed as a sequence of Gaussian pulses. The characteristics of these pulses, *e.g.*, peak amplitude and nominal duration, are functions of the physical properties of the obstacle. Therefore the reflected electromagnetic wave from a given obstacle is unique, from where the signature is obtained.

In [14], the UWB (Ultra Wide Band) technology has been used to create obstacle signatures. An FPGA-based hardware

accelerator has been proposed to meet the high real-time constraint due to the correlation computation. Our proposal is also based on the utilization of UWB radar to build an ORS system. As stated, the outputs of the ORS are sent to the MTT block to optimize the overall computation.

The general outline of data flow processing is presented in Fig. 3. As described, the recognition system consists of two key steps: 1). Radar detection and 2). Signature identification.

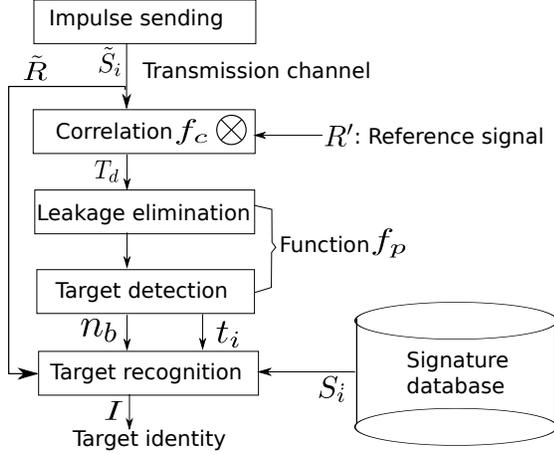


Fig. 3. General outline of data flow processing for signature identification

#### 1) Radar detection

The radar transmitter sends periodically an impulse signal  $\tilde{S}$ . The reflected signal  $\tilde{R}$  is modeled by a vector containing  $\tilde{N}$  samples:  $\tilde{R} = \{\tilde{r}_0, \dots, \tilde{r}_{\tilde{N}-1}\}$ . Similarly, the reference signal  $R'$  that corresponds to the reflected wave as transmitting an impulse signal to the receiver is represented by the  $\tilde{M}$ -element vector  $R' = \{r'_0, \dots, r'_{\tilde{M}-1}\}$ .

At the reception, a correlation between both the signals  $\tilde{R}$  and  $R'$  is performed to determine the respective distances to the obstacles. Mathematically, the correlation function  $f_c$  is expressed by Equation (1).

$$T_d = f_c(\tilde{R} \otimes R') = \sum_{i=0}^{\tilde{N}-1} \sum_{j=0}^{\tilde{M}-1} (\tilde{r}_{i+j} \times r'_j) \quad (1)$$

The obstacle distances are respectively calculated by checking the peak positions in the result  $T_d$ . In addition, the leakage elimination in Fig. 3 is to deal with the leakage transmission between both the transmission and reception antennas.

#### 2) Signature identification

The obstacle signature is identified by comparing the obstacle to a set of candidate signatures, denoted by  $S = \{s_0, \dots, s_{\tilde{K}-1}\}$  where the candidate  $s_i$  is itself a vector composed of  $\tilde{M}$  samples, as the reference signal  $R'$ . Similarly, this comparison is realized by the convolution equation. Hence, for obstacle  $k$ , the relevant samples  $r_k$  are successively correlated with the

candidate signatures  $s_i$ , as expressed in Equation (2).

$$V_i = f_c(r_k \otimes s_i) \text{ for } k \subseteq O_k \quad (2)$$

where the symbol  $O_k$  related to the  $k^{\text{th}}$  peak position represents the sub-set of sample subscripts in the received signal  $\tilde{R}$ . The signature pointed by the subscript  $i$  is selected in such a way that  $V_i = \max\{V\}$ .

### IV. MTT SYSTEM ARCHITECTURE

In this section, we first present an architecture overview of the MTT system. Secondly, we discuss in detail the obstacle signature application in the MTT system. Finally, the Munkres algorithm, *i.e.*, the assignment solver, is briefly reviewed.

#### A. MTT functional architecture

The MTT system tracks targets by consistently processing the observation data in three iterative stages, as shown in Fig.4.

- 1) Observation: The distance and angle values are given during a radar scan.
- 2) Data Association: In the *Obstacle to Track Assignment* block, the observations are mapped respectively to the tracked targets. The *Track Maintenance* block initiates new tracks or deletes existing ones when needed. The *Gate Computation* block defines the gate parameters from around the predictions [11].
- 3) Filtering and Prediction: The filtering block estimates the current position and predicts the next position of each target.

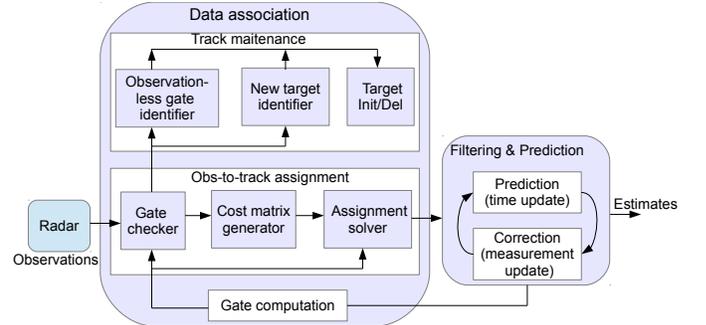


Fig. 4. Functional block architecture of the MTT application

#### B. MTT system modeling with signatures

The signature information helps to improve the DAS accuracy performance. We propose to introduce the obstacle signature  $S$  inside the MTT computation. It is then a must to take into account the signature information in all the MTT steps. In this case, the measurement and prediction are respectively expressed by the vectors  $V_m$  and  $V_p$  in Equation (3):

$$V_m = \begin{pmatrix} D \\ A \\ S \end{pmatrix} \text{ and } V_p = \begin{pmatrix} D \\ V \\ A \\ V_a \\ S \end{pmatrix} \quad (3)$$

where the symbols  $D$ ,  $A$ ,  $S$ ,  $V$ , and  $V_a$  correspond to distance, angle, signature, velocity and angular velocity, respectively.

### C. Cost matrix generation

Considering  $N$  tracks and  $N$  measurements during a scan, the problem that should be resolved is to establish the one-to-one relationship between the tracks and the measurements. The costs of the possible assignments are calculated in a 2D cost matrix of size  $N \times N$ . Mathematically, the assignment cost for the  $i^{th}$  measurement to the  $j^{th}$  track is expressed by Equation (4) :

$$c_{ij} = \frac{[\tilde{y}_{ijd} \ \tilde{y}_{ij\theta}] \begin{bmatrix} \tilde{p}_{jd \rightarrow \theta} + r_\theta & -\tilde{p}_{jd \rightarrow \theta} \\ -\tilde{p}_{j\theta \rightarrow d} & \tilde{p}_{j\theta \rightarrow d} + r_d \end{bmatrix} \begin{bmatrix} \tilde{y}_{ijd} \\ \tilde{y}_{ij\theta} \end{bmatrix}}{(\tilde{p}_{jd \rightarrow d} + r_d)(\tilde{p}_{j\theta \rightarrow \theta} + r_\theta) - \tilde{p}_{jd \rightarrow d}\tilde{p}_{j\theta \rightarrow \theta}} \quad (4)$$

where the symbols are defined as follows:

- The differences between measurements and predictions in distance  $d$  and angle  $\theta$  are respectively represented by  $\tilde{y}_d, \tilde{y}_\theta$ .
- The relationship  $\tilde{p}_{jd \rightarrow \theta}$  means the covariance  $\tilde{p}$  between distance and angle for the  $j^{th}$  track. The covariance progressively evolves following the tracking iteration. As indicated, each track is associated to a covariance matrix.
- Similarly, the representation  $r_d, r_\theta$  is for the constant transmission variance  $r$  related to the distance and the angle, respectively.

### D. Munkres algorithm

The Munkres algorithm is well adapted to the MTT application to resolve the assignment problem [11]. The principle is to establish the optimal association between the  $N$  measurements and the  $N$  predictions by minimizing the total cost. For this purpose, an iterative process is executed until the total assignment cost is not improved no longer. Moreover, a unique measurement can only be assigned to one track (prediction). The description of this algorithm is beyond the scope of this paper. The interested readers are referred to the article [15] for details.

As an example, giving a 2D cost matrix  $C$ :

$$C = \begin{pmatrix} 0 & 2 & 5 & 3 \\ 1 & 0 & 6 & 2 \\ 6 & 7 & 0 & 4 \\ 9 & 4 & 2 & 0 \end{pmatrix}$$

the Munkres algorithm generates a corresponding binary mask  $\tilde{M}$ :

$$\tilde{M} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (5)$$

where the binary value 1 represents the coupled assignment between the relevant measurement (line) and the track (column), 0 on the contrary. The iterative researching process is stopped if the following condition is verified:

$$\begin{cases} \sum \tilde{M}_{ij} = 1 & \forall i \in (0 \cdots N - 1) \\ \sum \tilde{M}_{ij} = 1 & \forall j \in (0 \cdots N - 1) \end{cases}$$

We obtain the minimal assignment cost of 0 with the mask given by Equation (5).

## V. GATE AND SIGNATURE CHECK

### A. Motivation

The Munkres algorithm basically relies on the cost matrix. For a given line  $i$  in the matrix  $C$ , if the polynomial  $\bar{S}$ , denoted by the expression :

$$\bar{S} = \sum_j (c_j - \min(c_j)) \rightarrow \infty, \quad (6)$$

is large enough, it can preliminary be determined with high probability that, the line element  $i$  is assigned to the column element  $j$ , in such a way that:

$$\tilde{m}_{ij} = 1 \text{ if } c_{ij} = \min(c_j). \quad (7)$$

The expression (7) can be considered as stop criteria during the iterative process. In this case, it is not necessary to run through all the steps required by the Munkres algorithm. Thanks to the stop criteria, the number of iterations can be reduced, thus the execution time. In the next of this section, we investigate two approaches making possible to fulfill both the equations (6) and (7).

### B. Gate Check

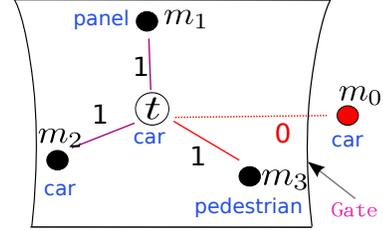


Fig. 5. Gate checking between 1 track (prediction) and 4 measurements.

The gate check method consists in calculating the binary mask matrix, which indicates the possible assignments of measurements and tracks. Only the measurements inside a specific window around each track are candidates for the assignment. The window is defined by the following rules:

$$\begin{cases} |\Delta_D| < K \times \sqrt{r_0 + p_0} \\ |\Delta_A| < K \times \sqrt{r_1 + p_1} \end{cases} \quad (8)$$

where the symbol  $\Delta$  represents the difference between the measurements and the track prediction. The subscripts  $D$  and  $A$  denote respectively distance and angle. The values  $r_i$  and  $p_i$  correspond respectively to the noise variance and the process co-variance and  $K$  is the gate coefficient. If the inequality (8) is not verified, the cost is set to an arbitrary large value.

The gate checking procedure for 4 measurements and 1 track is illustrated in Fig. 5 where possible assignments are marked by ones and others by zeros. Thanks to the gate checking, the measurement  $m_0$  is first eliminated due to its exceeded cost outside the gate area.

### C. Signature Check

As evoked, the radar measurement signal contains the obstacle signature information such as panel, car, pedestrian etc. At first, we consider only the binary signature identification, which implies that each obstacle does not have more than one signature. In other words, if the measurement  $m_i$  does not have the same signature to the track  $t$ , it means that the possibility to link these two elements is equal to 0. The above assumption leads to the signature check, which consists in comparing the signature identity as performing the assignment operation.

As shown in Fig. 5, the target track has the signature “car”. In this case, only the assignment  $t \rightarrow m_2$  is possible after applying the signature check, because both of them have the same identity. Similarly to the gate checking, the cost values of the non-possible assignments are equal to a large number. Again, the computational complexity is reduced because these non-possible assignments are with no needs to run the cost generation formula (see Equation (4)).

## VI. REAL-TIME PROFILE ANALYSIS

In this section, we first perform the time profiling by running the MTT application code on a Microblaze processor. Secondly, the time impact due to the obstacle signature utilization in the MTT system is analyzed.

### A. Time distribution of the MTT system

The time profiling of the ORS-MTT system is done through a Microblaze-based architecture. The architecture is implemented on the Xilinx ML605 kit board, on which an FPGA circuit Virtex 6 is included [16]. The relevant Xilinx embedded development tools are consequently used.

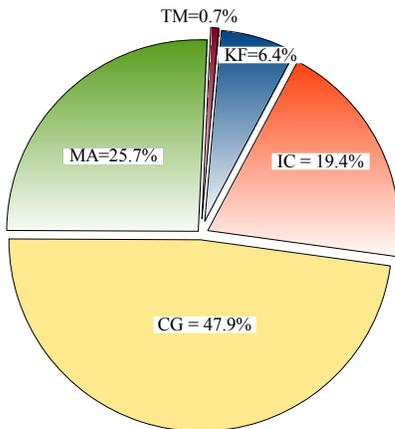


Fig. 6. Time distribution in the MTT system. The abbreviations are defined as follows: KF for Kalman Filter, IC for Innovation Computing, MA for Munkres Algorithm, TM for Track Maintenance and CG for Cost Generation.

We consider the MTT application using integer values for 1000 scans and a maximum of 20 obstacles. In Fig. 6, the execution time of the Kalman Filter (KF) corresponds to the processing of only one obstacle. It is shown that the most time consumed component functions are the Cost Generation (CG) and the Munkres Algorithm (MA), of which the execution

time accounts respectively for 25.7% and 47.9%. The time required by Innovation Computing (IC) accounts for 19.4%. As conclusion, the complexity of the functions CG and MA is relatively higher than the others.

According to Equation (4), the complexity of cost generation is equal to  $N^2$  operation units, *i.e.*  $\mathcal{O}(N^2)$ . The operation unit here includes multiplications, additions and division. For the Microblaze [17] processor, an addition and a multiplication needs respectively 1 and 3 clock cycles for integer values, whereas a division requires up to 34 clock cycles. On the other hand, the Munkres algorithm exposes an iterative process to obtain the optimal solution. The number of iterations might vary in function of the cost matrix. This explains that the cost generation and the Munkres algorithm need a relatively long time to execute.

### B. Execution Time Analysis

Two key factors are to be taken into account for the execution time: the number of obstacles  $N$  and the gate coefficient  $K$  (see Equation (8)). The first factor determines the computational complexity of generating the cost matrix  $C$ . The second one defines the size of the gate area, *i.e.*, the number of gates available for a possible assignment. With regards to the impact on the execution time, three cases are to be investigated for the overall MTT system: 1). Without gate check nor signature check, 2). With gate check only and 3). With gate and signature checks.

- 1) Without gate check nor signature check. As no checks are considered inside the application process, the cost matrix is to be generated with the highest complexity, *i.e.*  $\mathcal{O}(N^2)$  operations. This is equivalent to affect an infinite value to the gate coefficient  $K$  ( $K = \infty$ ). Hence, the application with no checks consumes the most important time. Instead of considering the total time consumed by the MTT application, it might be of interest to count merely the impacted part: the Cost Generation (CG) function and the Munkres Algorithm (MA). According to the execution times given in Fig. 7 (see  $K = \infty$ ), where the time basis in the vertical axis is changed to the sum of MA and CG times, we obtain a time of 15 milliseconds for 20 obstacle and 1 millisecond for 5 obstacles. This time consumption increases when the matrix size  $N$  increases.
- 2) With gate check only. The cost generation outside the gate area can be omitted thanks to the gate check. The gate size is determined by the gate coefficient  $K$ . Thus, the parameter  $K$  jointly with the number of obstacles  $N$  plays a primordial role to reduce the time consumption. According to Fig. 7, which depicts the time profile in function of the parameters  $N$  and  $K$ , we obtain a time of 14 milliseconds for the gate coefficient  $K$  equal to 30 and  $N$  to 20. This value changes to 11 milliseconds when  $K$  is equal to 3 for the same number of obstacles. Additionally, the gate coefficient  $K$  does not have a significant impact on the execution time when the number of obstacle  $N$  is small, *e.g.*  $N = 5$ .

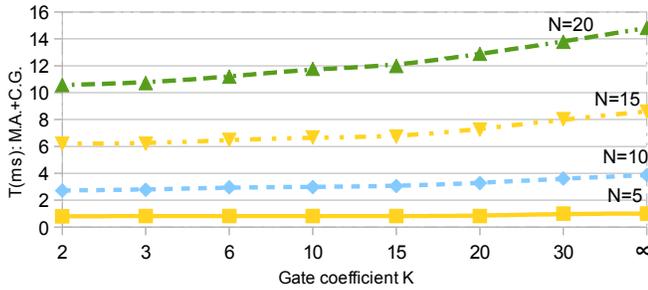


Fig. 7. Execution time (in mSec) of the MTT system with gate check for the Munkres algorithm (MA) and the cost generation (CG) within the Data Association block. The number of obstacles ( $N$ ) varies between 5 and 20.

3) With gate and signature checks. We first introduce the recognition ratio  $R$  defined by the expression (9):

$$R = \frac{N_c}{N} \quad (9)$$

where the symbol  $N_c$  denotes the number of unique signatures included inside a radar scan and  $N$  is the total number of obstacles. For instance, if there are 3 identified obstacles, let's say 2 cars, 1 pedestrian. The recognition ratio is equal to  $\frac{2}{20}$  because of the two unique signatures: car and pedestrian.

The execution time in function of the recognition ratio  $R$  and the parameter  $K$  is shown in Fig. 8. We obtain an execution time of 11 milliseconds for  $R = \frac{1}{20}$  and  $k = 3$ . This value decreases to 7.5 milliseconds when the ratio  $R$  raises up to  $\frac{20}{20}$ , from where a speed up of 32% is obtained. Regardless of the value of the parameter  $K$ , the execution time converges to the minimum time value of 7.5 milliseconds with the recognition ratio equal to  $\frac{20}{20}$ , i.e.  $R = 1$ . It indicates that the minimum execution time is achieved when no identical obstacles are observed within one radar scan.

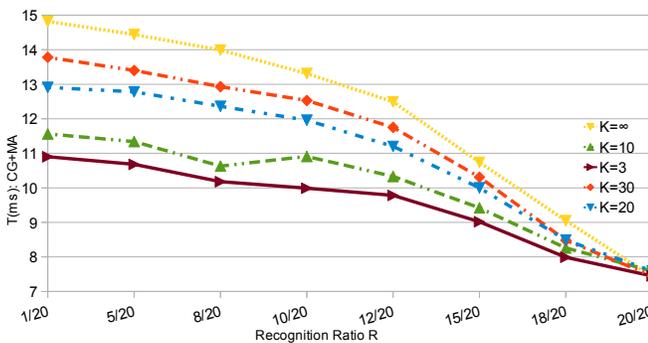


Fig. 8. Execution time (in mSec) with gate and signature checks for Munkres Algorithm (MA)+Cost Generation (CG) for different values of gate coefficient  $K$ . The number of obstacles is set to 20 and the number of unique signatures  $N_c$  varies between 1 and 20.

## VII. CONCLUSION

The radar signature utilization in automotive MTT system has been presented in this paper. Thanks to the signature utilization, it is possible to categorize the obstacles, to recognize

the degree of dangerousness and to reduce the false alarms. On the other hand, the application of signatures eases the cost matrix generation and thus increases the timing efficiency. The time profile has been investigated over a soft-core processor Microblaze. Compared to the standard MTT system, the necessary time is dropped down from 11 milliseconds to 7.5 milliseconds by jointly using the gate and signature checks.

## ACKNOWLEDGEMENT

We are indebted to Yassin El-Hillali and Atika Rivenq for the extensive support they provided. This research project is supported by the International Campus on Safety and Intermodality in Transportation (CISIT in French).

## REFERENCES

- [1] H. Winner, S. Witte, W. Uhler, and B. Lichtenberg, "Adaptive cruise control system: Aspects and development trends," *SAE Paper 961010*, no. DOI: 10.4271/961010, 1996.
- [2] I. Masaki, "ASIC approaches for vision-based vehicle guidance," *IEICE Transactions on Electronics*, vol. E76-C, no. 12, pp. 1735–1743, Dec. 1993.
- [3] M. Valera and S. A. Velastin, "Intelligent distributed surveillance systems: A review," *IEE Proceedings - Vision, Image and Signal Processing*, vol. 152, no. 2, pp. 192 – 204, 8 April 2005.
- [4] J. Saad, A. Baghdadi, and F. Bodereau, "FPGA-based radar signal processing for automotive driver assistance system," in *IEEE/IFIP International Symposium on Rapid System Prototyping*, Paris, 23-26 June 2009, pp. 196 – 199.
- [5] J. McCall and M. Trivedi, "Video-based lane estimation and tracking for driver assistance: survey, system, and evaluation," *IEEE Transactions on Intelligent Transportation Systems*, pp. 20–37, 2006.
- [6] M. Wada, S. Y. Kang, and H. Hashimoto, "Development of advanced parking assistance system," *IEEE Transactions on Industrial Electronics*, vol. 50, pp. 4–17, 2004.
- [7] A. Vahidi and A. Eskandarian, "Research advances in intelligent collision avoidance and adaptive cruise control," *IEEE Transactions on Intelligent Transportation Systems*, vol. 4, no. 3, pp. 143–153, 2003.
- [8] H. Hiroto, "Automotive image recognition processor imapar," *NEC technical journal ISSN 1880-5884*, vol. 1, no. 5, pp. 24–28, Dec. 2006.
- [9] G. P. Stein, E. Rushinek, G. Hayun, and A. Shashua, "A computer vision system on a chip: a case study from the automotive domain," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, San Diego, CA, USA, 20-26 June 2005, pp. 130 – 130.
- [10] C. Claus, W. Stechele, and A. Herkersdorf, "Autovision a run-time reconfigurable mp soc architecture for future driver assistance systems," *Information Technology*, vol. 49, no. 3, pp. 181 – 187, 2007.
- [11] J. Khan, S. Niar, A. Menhajl, Y. Elhillali, and J. L. Dekeyser, "An mp soc architecture for the multiple target tracking application in driver assistance system," in *The 19th IEEE International Conference on Application-Specific Systems, Architectures and Processors*, Leuven, Belgium, July 2008, pp. 126 – 131.
- [12] N. Harb, M. Saghir, and S. Niar, "A dynamically reconfigurable kalman filtering block for an automotive multiple target tracking system," in *The 6th HiPEAC Workshop on Reconfigurable Computing*, Paris, Jan. 2012.
- [13] M. Hussain, "Principles of high-resolution radar based on nonsinusoidal waves. iii. radar-target reflectivity model," *IEEE Transactions on Electromagnetic Compatibility*, vol. 32, issue 2, no. 3, pp. 144 – 152, 2002.
- [14] H. Liu, S. Niar, Y. El-Hillali, and A. Rivenq, "Embedded architecture with hardware accelerator for target recognition in driver assistance system," *Newsletter, ACM SIGARCH Computer Architecture News*, vol. 39, issue 4, 2011.
- [15] T. Gottschalk, "Concurrent implementation of munkres algorithm," in *Proceedings of the Fifth Distributed Memory Computing Conference*, no. DOI: 10.1109/DMCC.1990.555361, Charleston, South Carolina, Apr. 1990, pp. 52–57.
- [16] Xilinx, "Xilinx virtex-6 fpga configuration user guide ug360 (v3.2)," November 2010.
- [17] —, "Xilinx microblaze processor reference guide, ug081 (v11.0)," *Embedded Development Kit EDK 12.1*, Sep. 14 2000.