Sensor networks for real-time vehicle tracking on road networks

Sensor Network for Real-Time Vehicle Tracking on Road Networks

Zoltan Papp[1], Joris Sijs[2], Michele Lagioia[3]

TNO Science and Industry
Delft, The Netherlands
[1] zoltan.papp@tno.nl, [2] joris.sijs@tno.nl, [3] michele.lagioia@tno.nl

***Abstract*—Due to the continuously increasing mobility demand the efficiency of the existing road infrastructure should be improved. Advanced control and management techniques offer promising, cost effective and environmentally friendly solutions to the problem. As traffic and vehicle control/management applications become more advanced they require more and more detailed and accurate real-time observations about the surrounding relevant world. The paper describes a road infrastructure based sensing and data interpretation system, which is scalable, robust and delivers the motion state estimation of the individual vehicles in real-time. The system enables a wide spectrum of intelligent transport system functionalities ranging from traffic management to fully autonomous driving. Distinguishing features of the solution that it does not require instrumentation on the vehicles to become detected and it is insensitive to ambient environmental conditions (weather, light, etc.). The paper describes the event driven object tracking algorithm, and its "mapping" to distributed computing platform resulting in an inherently robust implementation.**

I. Introduction

It is estimated that, due to further demand for mobility, before 2020 the traffic will increase more than 40%. Without drastic measures this will result in serious problems in terms of society, safety, economics and environment [1]. There are a number of proposals/alternatives to answer these challenges, ranging from advanced traffic management to fully automated cooperative driving. Though these management/control applications cover a wide range of functionalities and temporal dynamics, they share a common property: in order for an application "to do the right thing at the right time" it should have a formal representation and understanding of the relevant surrounding world [2]. Consequently the advanced management/control functionalities pose extreme challenges for sensing and data interpretation [3], [4]. The solutions available differ in the sensing principle (e.g. video, radar, GPS), the temporal properties (e.g. sampling time, delay, etc.) resulting in differences in granularity (e.g. measuring traffic streams or individual vehicles), accuracy and dependability features. The systems providing traffic and/or motion state information fall into two categories. The first category uses on-board sensors and car-to-car communication. In this case the individual vehicles estimate their own motion state and via communication links they share this information with others. The advanced control/management services require relatively high penetration ratio of these intelligent vehicles on the road [4], [5]. Certain safety critical applications do not allow "invisible" vehicles, i.e. 100% penetration ratio is a condition for safety - which makes the introduction virtually impossible. The second category is often called intelligent infrastructure, in which the infrastructure carries out the observation and provides the processed data set for traffic management and (via wireless links) onboard services. This category has lower "introduction threshold" and better dependability features. The loop detectors, video based monitoring are typical instances in this category. Recently the wireless sensor network technology gave a new impulse to the research on novel sensing, montoring and control techniques. [6] describes a vehicle and traffic monitoring solution, which uses both vehicle and infrastructure components. [7] reports a work on road embedded wireless sensor network, but it focuses on traffic characteristics measurements and vehicle classification.

In this paper an infrastructure based sensing and data interpretation solution is presented, which exhibit the following features:

- scalable in space, resolution and accuracy,
- real-time observation stream,
- inherently dependable,
- no need of any instrumentation of the participating vehicles.

The scalability in resolution and accuracy allows for covering wide application spectrum ranging from traffic parameter measurements (e.g. average speed, traffic density) to tracking of individual objects (vehicles) on the road.

The paper is structured as follows. The next section describes the "big picture", i.e. the main concepts behind the proposed solution and identifies the main functional components of the system. In Section III the signal processing aspect is considered. Section IV describes how the functional architecture is mapped into a distributed computing platform. The following section summarizes the main features of the test-bed, which is used to evaluate algorithmic and architectural alternatives under "laboratory conditions".

II. The approach

The real-time object tracking system presented uses road embedded sensors for vehicle detection and based on the time of detections it estimates the motion state of the individual vehicles. Considering the special requirements of the application domain and the various constraints of the deployment (see more details in the system architecture section) a discrete

978-1-4244-3518-0/09/$25.00

observation scheme used (Fig. 1). Giant magneto-resistive

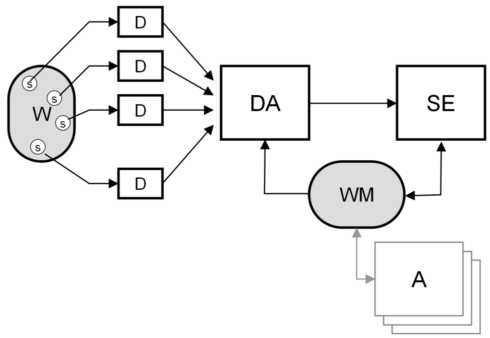

Fig. 1. The functional scheme of the object tracking

(GMR) sensors (S in the figure) are used to measure the changes in the Earth magnetic field caused by the presence of a ferromagnetic object (vehicle). The sensory signal is fed into the detection block (D), which delivers the binary valued observation events (object arrived/left, time of event, detection position) reflecting the state changes in the observed physical phenomenon in the real world (W). The data association (DA) identifies[1] the most probable source of the detection based on the currently known state of the world as represented in the world model (WM). Having the object identified and the position and pose calculated by the data association, the observations are passed to the state estimator (SE), which updates the object's state in the world model. Applications (A) use the real-time world model to implement control, deliver services, etc. The density of the sensor deployment determines the uncertainties of the object identification and motion state estimation. The next section describes the method developed for object tracking and its statistical properties. The section on system architecture addresses the scalability and real-time issues.

III. Object tracking with discrete observations

The aim is to track multiple objects moving on a 2D-plane based on discrete detections on that plane. To do that state estimator is used in which each object i has a corresponding state vector $s^i := (x^i, y^i, \theta^i, \dot{x}^i, \dot{y}^i)^T$ and process noise $w^i := (\ddot{x}^i, \ddot{y}^i, \dot{\theta}^i)^T$. Further $o^i := (x^i, y^i)^T$ and θ^i represent the i^{th} object's position vector and orientation respectively, as shown in Fig. 2. To have a stable state estimator, the measurement vector m^i is defined as the object's pose, i.e. $m^i := \left((o^i)^T, \theta^i\right)^T$. Thus, if $G(x, \mu, U)$ denotes the Gaussian function of $x \in \mathbb{R}^n$ defined as $((2\pi)^n det(U))^{-0.5} e^{(x-\mu)U^{-1}(x-\mu)}$, the following state-space model[2] describes the dynamics of object i (with a constant sampling time t_s):

$$s^i[k] := As^i[k-1] + Bw^i[k], \tag{1a}$$

$$m^i[k] := \begin{pmatrix} I^{3\times3} & \underline{0} \end{pmatrix} s^i[k] + v^i[k], \tag{1b}$$

$$\text{with } p(w^i[k]) := G(w^i[k], 0, Q^i) \tag{1c}$$

and

$$A := \begin{pmatrix} 1 & 0 & 0 & t_s & 0 \\ 0 & 1 & 0 & 0 & t_s \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}, \quad B := \begin{pmatrix} 0.5t_s^2 & 0 & 0 \\ 0 & 0.5t_s^2 & 0 \\ 0 & 0 & 1 \\ t_s & 0 & 0 \\ 0 & t_s & 0 \end{pmatrix}.$$

Whenever a crossing vehicle is detected by a particular sensor, the sensor sends its position (i.e. the detection position), denoted with u, to the data association unit. The association is carried out by using the Gating and Nearest Neighbor [8] method. However, it cannot be assumed that the detection will be perfect, i.e. the sensor triggers exactly when the object's boundary crosses the detection point u. In reality a detection will be affected by noise. This is modeled by redefining the each position of a detection-sensor as the random vector $\tilde{u}$ having a mean u and covariance ϵI:

$$p(\tilde{u}) := G(\tilde{u}, u, \epsilon I). \tag{2}$$

Fig. 2 shows an example in which object i is detected by multiple detection points having a covariance ϵ. Notice that in case of a detection, the object's interior must cover a certain area around a detection point.

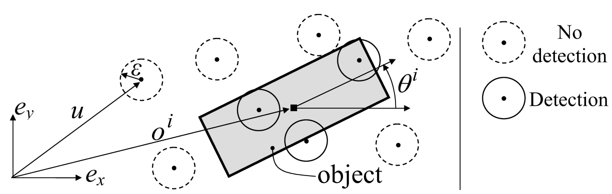

Fig. 2. Detection of object i at multiple detection points

After associating the detections, the object tracking invokes two functionalities. First a pose-estimating function is executed, which estimates $m^i[k]$ based on the current set of detections of the vehicle. The output of this function is a PDF $p(m^i_k)$ described as a sum of Gaussians. This PDF is then used in the state-estimator function, which calculates the PDF $p(s^i_k)$ as a single Gaussian based on all previous estimated measurements $m^i[0]$ until $m^i[k]$. In order to simplify the notation we will drop the superscript i and for the pose-estimator also the sample instant k.

1) Pose-estimation: Let us assume that the current set of detections of a vehicle are collected in the set $Z := \{u_1, u_2, ..., u_n\}$. The part of the road surface that is covered by the vehicle, denoted by its "shadow" $O(o, \theta) \in \mathbb{R}^2$, is assumed to be rectangular. As the vehicle moves, so does its "shadow", due to which $O(o, \theta)$ depends on the position and orientation of the vehicle. The length and width of the vehicle's "shadow" are assumed to be constants and known.

[1]Identification leaves the real objects anonymous. A label is attached to each identified objects automatically but this label serves merely internal "administrative" purposes to support the tracking process.

[2]The composition of the state vector and the corresponding system model depend on the particular application case addressed. For example in typical highway applications the orientation of the objects (vehicles) is not estimated because it is mainly determined by the road geometry - which also greatly simplifies the pose estimation procedure.

The principle of the pose-estimator is as follows: First the estimation the position o of the vehicle is calculated as the Gaussian $G(o, o_l, V_l)$, given the set of detection sensors and given a certain orientation, i.e. $p(o|Z, \theta_l) = G(o, o_l, V_l)$. Then this PDF is calculated for a total of L of orientations, i.e. $\theta_l = l2\pi/L$, after which the final PDF of the measurement vector becomes: $p(m|Z) = \sum_{l=1}^{L} \beta_l G(o, o_l, V_l)$.

For estimating o given Z notice that, in case the vehicle is detected at $\tilde{u}_j$, then $O(\tilde{u}_j, \theta_l)$ is the set of all possible positions o of the vehicle, given a certain orientation θ_l. In case more sensors detection the vehicle, then one can define the set $M(\theta_l) \in \mathbb{R}^2$ as the area which is covered by all the sets $O(\tilde{u}_j, \theta_l)$ for each detection-position u_j. As such it follows that $M(\theta_l) := \cap_{j \in \{1,\ldots,n\}} O(\tilde{u}_j, \theta_l)$, which is also shown in Fig. 3. Notice that $M(\theta_l)$ is the set of all possible positions o, at a certain orientation θ_l, given all sensors of Z that detect the vehicle. Therefore, in case c_M is a constant equal to the total surface of $M(\theta_l)$ and c_O is a constant equal to the total surface of $O(o, \theta_l)$, then $p(o|u_j, \theta_l)$ and $p(o|Z, \theta_l)$ can be modeled as:

$$
\begin{aligned}
p(o|\tilde{u}_j, \theta_l) &= \begin{cases} 0 & \text{if} \quad o \notin O(\tilde{u}_j, \theta_l) \\ c_O^{-1} & \text{if} \quad o \in O(\tilde{u}_j, \theta_l) \end{cases} \\
p(o|Z, \theta_l) &= \begin{cases} 0 & \text{if} \quad o \notin M(\theta_l) \\ c_M^{-1} & \text{if} \quad o \in M(\theta_l) \end{cases}
\end{aligned}
\tag{3}
$$

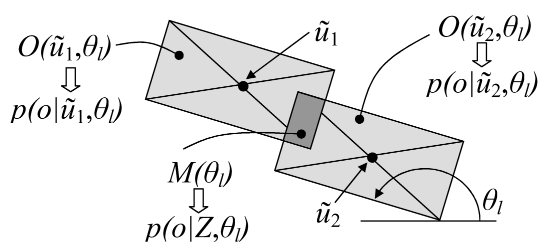

Fig. 3. Determining the set of possible position considering the different position of a detection $\tilde{u}_j$

The PDFs of (3) are difficult to deal with in an algorithm. Therefore both are approximated as a summation of Gaussians. For that, let us define that the PDF of a single detection is approximated by M Gaussians, i.e. $p(o|\tilde{u}_j, \theta_l) := \sum_{m=1}^{M} \frac{1}{M} G(o, o_m, V)$. Notice however, that $\tilde{u}_j$ is in fact an estimate of the real position at which the vehicle was detected, i.e. $G(\tilde{u}_j, u_j, \epsilon I)$. Therefore, $p(o|u_j, \theta_l)$ equals the convolution integral between $p(o|\tilde{u}_j, \theta_l)$ and $G(\tilde{u}_j, u_j, \epsilon I)$:

$$
p(o|u_j, \theta_l) = \sum_{m=1}^{M} \frac{1}{M} G(o, o_m, V + \epsilon I). \tag{4}
$$

Notice that the position's PDF based on all detection, i.e. $p(o|Z, \theta_l)$, is calculated by multiplying the PDFs of each individual detection:

$$
\begin{aligned}
p(o|Z, \theta_l) &= \prod_{j \in Z} p(o|u_j, \theta_l), \\
&= \prod_{j \in Z} \sum_{m=1}^{M} \frac{1}{M} G(o, o_m, V + \epsilon I).
\end{aligned}
\tag{5}
$$

The state-estimator requires the PDF $p(m|Z)$, which is defined as the summation of each PDF $p(o|Z, \theta_l)$ at the different orientations. However, to keep a computation tractable algorithm the PDF of (5) is approximated as a single Gaussian; $p(o|Z, \theta_l) \approx G(o, o_l, V_l)$ with a total probability of β_l. Therefore the PDF $p(m|Z)$ results in a summation of Gaussians:

$$
p(m|Z) \approx \sum_{l=1}^{L} \frac{\beta_l}{\sum_{l=1}^{L} \beta_l} G(o, o_l, V_l). \tag{6}
$$

2) State-estimation: The state-estimator is based on the Kalman filter [9] and the Gaussian Sum Filter [10], [11], because $p(m[k]|Z[k])$ is a summation of Gaussians. These types of estimators are well described, thus here only the resulting algorithm is presented. In this algorithm $x^-[k]$ and $P^-[k]$ represents the predicted state-vector and covariance-matrix of the k^{th} sample instant respectively. Further, $x[k]$ and $P[k]$ represents the updated state-vector and error-covariance matrix of the k^{th} sample instant respectively:

prediction-step:

$$
\begin{aligned}
&x^-[k] = Ax^-[k-1], \\
&P^-[k] = AP^-[k-1]A^T + BQB^T,
\end{aligned}
\tag{7a}
$$

measurement-update:

$$
\begin{aligned}
&K_l[k] = P^-[k]C^T \left(CP^-[k]C^T + V_l[k]\right)^{-1}, \\
&x_l[k] = x^-[k] + K_l[k]\left(m_l[k] + Cx^-[k]\right), \\
&P_l[k] = \left(I + K_l[k]C\right)P^-[k].
\end{aligned}
\tag{7b}
$$

state-estimates:

$$
\begin{aligned}
&x[k] = \sum_{l=1}^{L} \frac{\beta_l[k]}{\sum_{l=1}^{L} \beta_l[k]} x_l[k], \\
&e[k] = \sum_{l=1}^{L} \frac{\beta_l[k]}{\sum_{l=1}^{L} \beta_l[k]} x[k] - x_l[k], \\
&P[k] = \sum_{l=1}^{L} \frac{\beta_l[k]}{\sum_{l=1}^{L} \beta_l[k]} \left(P_l[k] + e[k]e^T[k]\right).
\end{aligned}
\tag{7c}
$$

In order to test the object-tracking algorithm, an example of parking lot of 50 by 50 meters is simulated. A total of 2500 detection sensors were used, i.e. a density of one sensor per square meter. The 4 vehicles are all rectangular shaped objects of 5 by 2 meters, maneuvering within the parking lot. The results of the individual tracks of each vehicle are presented in Fig. 4. The real track of each vehicle is plotted in a solid line while a number of position of its estimated track are represented with a unique symbol; '$\circ$', '$\square$', '$*$' and '∇'.

IV. The system architecture

The system architecture gives the framework for implementing the signal flow on a select computing platform. The implementation should satisfy a number of non-functional requirements: for the targeted real-time vehicle tracking application domain the most important requirements are as

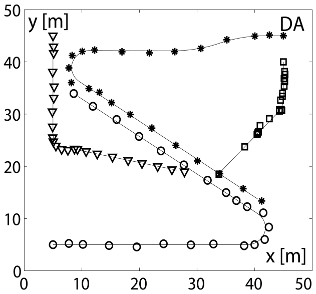

Fig. 4. Tracking simulation results

follows[3]:

- *Update rate:* The world model update should be carried out under real-time constraints. The system architecture should be "scalable in time", i.e. the system architecture should allow various trade-offs to customize the update rate to the particular need of the application addressed.
- *Spatial scalability:* The vehicle tracking system should be able to cover arbitrarily large areas without any architectural or component level adjustments (enabling "on-the-fly" extensions).
- *Scalability in accuracy:* The accuracy of the motion state estimation depends on the density of the detector deployment. The system architecture should allow for variable density of detector installations and "on-the-fly" introduction of new detection points.
- *Robustness, graceful degradation:* It is expected that during the lifetime of the system a significant portion of the detectors would fail. The system should tolerate the loss of detectors and should be able to deliver its primary function. Beside the motion state information its uncertainty also should be provided.
- *Power efficiency:* The power source for the road-embedded detectors is seriously constrained. Currently (and in the coming years) only the combination of (rechargeable) battery and power harvesting seems to be a feasible solution. The required long operational life can only be achieved via power efficient architecture, hardware and software design.

In order to provide both spatial and accuracy scalability the distributed implementation of the signal flow is the only feasible alternative. In the literature great variety of system architectures were reported for object tracking problems [12], [13]. In our case the stringent real-time and power constraints make those architecture alternatives feasible, which

- minimize communication,
- allocate the computationally demanding functionalities on (powered) roadside nodes and
- rely on single-hop networking[4].

[3]This is not a full list of non-functional requirements, but these are the most critical ones, which primarily shaped the system architecture.

[4]A number of design alternatives were modeled and evaluated analytically and in (discrete event) simulations - the paper describes only the one, which was selected for implementation.

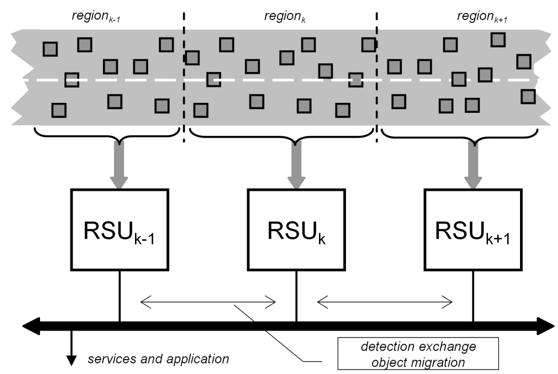

Fig. 5. The distributed processing architecture

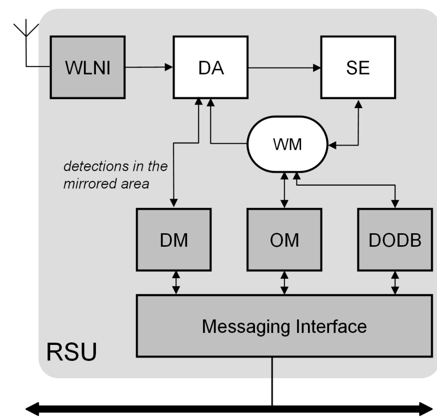

Fig. 6. The roadside processing unit

Fig. 5 and Fig. 6 show the distributed functional and communication architecture of the vehicle tracking system. Each unit embedded in the road (dark grey squares in Fig. 5) hosts one single instance of sensing and detection (D) functionalities. The detection events (detection value, sensor id) are communicated with the roadside units (RSU) wirelessly. Each RSU is responsible for covering a predefined region of the road, i.e. it receives the detection messages from those units, which belong to its assigned region and estimates the motion state of the objects staying in the region[5]. Fig. 6 details the internal of an RSU. The gray blocks represent those "extension functionalities", which connect the data interpretation to the sensing and the neighboring RSUs. The detection events are received via wireless network interface (WLNI). The data association (DA) and state estimation (SE) stages build the world model (WM) describing the objects contained by the region. The "ownership" of an object is defined by the position of its center point with respect to the region: an RSU "owns" an object if its center point falls into the region covered by the RSU. RSU carries out state estimation and maintains the world model for all objects it owns. At region borders relevant detections may happen in the neighboring region (i.e.

[5]The method and its implementation can easily be extended to handle arbitrary disjunctive regions.

handled by another RSU). In order to carry out the data association these detections should be shared by the regions involved (see also Fig. 7). The detection mirroring (DM) and object migration (OM) blocks implement the detection sharing and object ownership management, respectively. The

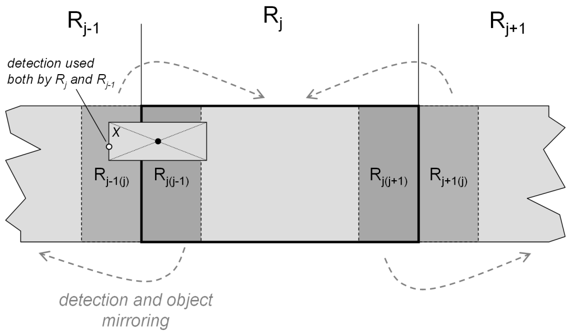

Fig. 7. Detection and object management

system architecture described has a number of advantageous implementational features:

- The road embedded components are extremely simple, consequently low cost, power efficient implementation is feasible.
- The road embedded components use one-way communication link to the associated RSU. It allows simple wireless communication design, which further decreases complexity and costs.
- The one-way communication simplifies the communication protocol design [14]. Because the data interpretation is resilient to message loss to great extent no attempt is made to avoid message collision and there is no recovery mechanism used[6].
- The event driven data interpretation scheme is economical both with respect to power consumption and message loss, consequently it enables higher detection density and accuracy at the same power consumption level.

V. Technology test-bed, current status

In order to support algorithms development, evaluation and testing a technology test-bed was developed.

The test-bed can be considered as a scaled down of a real-life system. Beneath a 1 m by 1.8 m oval track (wooden surface) 150 sensors/detectors are placed, which wirelessly communicate with maximum four RSU receivers. The RSU receivers are connected to desktop PCs, which run the RSU functionalities. On one PC many RSU functionalities can be instantiated: these virtual RSUs use the Messaging Interface block for communication. Appr. 5 cm by 8 cm model cars (or other ferromagnetic objects) can be used as objects on the surface to be tracked. The main features of the test-bed:

- The test-bed implements the complete architecture of real-life systems.
- The sensor nodes use the same detection and communication protocols as the real-life deployment does.
- The data association and state estimation algorithms running in the RSUs are different of the real-life versions only in parameters.
- The test-bed allows for injecting failures thus enables algorithms tests with respect to robustness.

The test-bed has certain limitations, too:

- The number of sensor are limited (physical constraints), thus the high-density real-life deployment (appr. 1 sensor/m^2) cannot be achieved. This limits the accuracy of the tracking and constraints the evaluation of the graceful degradation properties.
- The "object signature" of the model cars (i.e. the analogue signal of the magneto-resistive sensor) can differ from that of real objects, consequently the detection algorithm testing is limited.
- The powering of the wireless nodes is wired. Obviously, the power harvesting is beyond the scope of the test-bed.

VI. Conclusions, future work

In the paper a distributed object tracking solution was presented, which is especially suitable for real-time vehicle tracking on road networks. The solution is scalable in space, in resolution, in accuracy and in robustness and does not require any instrumentation on the vehicles (for monitoring). The system builds a real-time vehicle map "on the roadside": the high accuracy real-time map, which identifies the individual vehicles opens up new possibilities for advanced monitoring and control. Via making the real-time world model available for the individual vehicles (through wireless infrastructure-to-vehicle communication) real-time closed loop vehicle control schemes can also be implemented. The road infrastructure based world model would act as an "intelligent sensor" for the vehicles extending their view and understanding of the surrounding. The same information infrastructure can also be used to represent and distribute other traffic related information (e.g. icy area, vehicle breakdown, dynamic speed limit, etc.).

In the immediate future a small scale pilot deployment on a multi-lane highway will be completed to test and tune the algorithms under highway traffic conditions. The road deformation/vibration driven power harvesting solution for powering the wireless sensor/detector nodes is under development. In parallel, emphasis is given to the novel traffic management related applications made possible by the real-time world model. Currently stand-still detection in emergency lanes, shock-wave damping in traffic congestions and model predictive control based dynamic speed advice are under development.

[6]Simulation experiments show that the worst case message loss is below 20% even in the case of very dense traffic driving beyond the safe speed range. This loss can easily be compensated by redundancy.

References

[1] http://www.foresight.gov.uk/OurWork/CompletedProjects/IIS/, "Foresight program." [Online]. Available: http://www.foresight.gov.uk/OurWork/CompletedProjects/IIS/

[2] R. Horowitz and P. Varaiya, "Control design of an automated highway system," in *Proceedings of the IEEE*, 2000, pp. 913–925.

[3] L. Baskar, B. De Schutter, and H. Hellendoorn, "Dynamic speed limits and on-ramp metering for IVHS using model predictive control," in *Proc. of the 11th Int. IEEE Conference on ITS*, Beijing, China, Oct. 2008, pp. 821–826.

[4] S. Halle, J. Laumonier, and B. Chaib-Draa, "A decentralized approach to collaborative driving coordination," in *Proc. of the 7th Int. IEEE Conference on ITS*, 2004, pp. 453–458.

[5] B. van Arem, C. J. G. van Driel, and R. Visser, "The impact of cooperative adaptive cruise control on traffic-flow characteristics," *Intell. Transp. Sys., IEEE Trans. on*, vol. 7, no. 4, pp. 429–436, 2006.

[6] V. Verma, R. Choudhari, S. Singh, A. Singh, T. Mishra, and P. Srivastava, "Intelligent transport management system using sensor networks," in *IEEE Intelligent Vehicles Symposium*, 2008.

[7] A. Knaian, "A wireless sensor network for smart roadbeds and intelligent transportation systems," 2000. [Online]. Available: citeseer.ist.psu.edu/knaian00wireless.html

[8] Y. Bar-Shalom and R. Li, *Multitarget-Multisensor Tracking: Principles and Techniques*. YBS, 1995.

[9] R. Kalman, "A new approach to linear filtering and prediction problems," *Transaction of the ASME Journal of Basic Engineering*, vol. 82, no. D, pp. 35–42, 1960.

[10] H. Sorenson and D. Alspach, "Recursive Bayesian estimation using Gaussian sums," *Automatica*, vol. 7, pp. 465–479, 1971.

[11] J. Sijs and M. Lazar, "On event based state estimation," in *Hybrid Systems: Computation and Control*, ser. Lecture Notes in Comp. Sci., vol. 5469. Springer Verlag, 2009, pp. 336–350.

[12] W.-C. P. Chih-Yu Lin and Y.-C. Tseng, "Efficient in-network moving object tracking in wireless sensor networks," *IEEE Trans. on Mobile Computing*, vol. 5, no. 8, pp. 1044–1056, 2006.

[13] T. He, S. Krishnamurthy, J. A. Stankovic, T. Abdelzaher, L. Luo, R. Stoleru, T. Yan, L. Gu, J. Hui, and B. Krogh, "Energy-efficient surveillance system using wireless sensor networks," in *Proc. of the 2nd Int. Conf. on Mobile Systems, Applications, and Services*. New York, NY, USA: ACM, 2004, pp. 270–283.

[14] S. C. Ergen and P. Varaiya, "Pedamacs: Power efficient and delay aware medium access protocol for sensor networks." *IEEE Trans. Mob. Comput.*, vol. 5, no. 7, pp. 920–930, 2006.