

Using Bayesian Programming for Multi-Sensor Multi-Target Tracking in Automotive Applications

Christophe Coué, Thierry Fraichard, Pierre Bessiere, Emmanuel Mazer

▶ **To cite this version:**

Christophe Coué, Thierry Fraichard, Pierre Bessiere, Emmanuel Mazer. Using Bayesian Programming for Multi-Sensor Multi-Target Tracking in Automotive Applications. –, 2003, France. hal-00068773

HAL Id: hal-00068773
https://hal.science/hal-00068773

Submitted on 17 Jun 2006

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Int. Conf. on Robotics and Automation
May 12-17, 2003. Taipai (Taiwan)

Using Bayesian Programming for Multi-Sensor Multi-Target Tracking in Automotive Applications

C. Coué, Th. Fraichard, P. Bessière and E. Mazer
Inria[a] Rhône-Alpes & Gravir[b]-CNRS[c]
655 av. de l'Europe, Montbonnot, 38334 St Ismier Cedex, France
Tel. +33 476 61 54 36 — Fax. +33 476 61 52 10
christophe.coue@inria.fr
http://www.inrialpes.fr/sharp

March 19, 2003

final version

Abstract

A prerequisite to the design of future Advanced Driver Assistance Systems for cars is a sensing system providing all the information required for high-level driving assistance tasks. Carsense is a European project whose purpose is to develop such a new sensing system. It will combine different sensors (laser, radar and video) and will rely on the fusion of the information coming from these sensors in order to achieve better accuracy, robustness and an increase of the information content. This paper demonstrates the interest of using probabilistic reasoning techniques to address this challenging multi-sensor data fusion problem. The approach used is called *Bayesian Programming*. It is a general approach based on an implementation of the Bayesian theory. It was introduced first to design robot control programs but its scope of application is much broader and it can be used whenever one has to deal with problems involving uncertain or incomplete knowledge.

Keywords — Bayesian reasonning, multi-sensor target tracking, data-fusion.

Acknowledgements — This work was partially supported by the French programme *"La Route Automatisée"* (http://www.lara.prd.fr/) and the European project IST-1999-12224 *"Sensing of Car Environment at Low Speed Driving"* (http://www.carsense.org).

[a]Institut National de Recherche en Informatique et en Automatique.
[b]Lab. Graphisme, Vision et Robotique.
[c]Centre National de la Recherche Scientifique.

Using Bayesian Programming for Multi-Sensor Multi-Target Tracking in Automotive Applications

C. Coué, Th. Fraichard, P. Bessière and E. Mazer
Inria Rhône-Alpes & Gravir -CNRS
http://www.inrialpes.fr/sharp

Abstract— **A prerequisite to the design of future Advanced Driver Assistance Systems for cars is a sensing system providing all the information required for high-level driving assistance tasks. Carsense is a European project whose purpose is to develop such a new sensing system. It will combine different sensors (laser, radar and video) and will rely on the fusion of the information coming from these sensors in order to achieve better accuracy, robustness and an increase of the information content. This paper demonstrates the interest of using probabilistic reasoning techniques to address this challenging multi-sensor data fusion problem. The approach used is called *Bayesian Programming*. It is a general approach based on an implementation of the Bayesian theory. It was introduced first to design robot control programs but its scope of application is much broader and it can be used whenever one has to deal with problems involving uncertain or incomplete knowledge.**

I. Introduction

Unlike regular cruise control systems, Adaptive Cruise Control (ACC) systems use a range sensor to regulate the speed of the car while ensuring collision avoidance with the vehicle in front. ACC systems were introduced on the automotive market in 1999. Since then, surveys and experimental assessments have demonstrated the interest for this kind of systems. They are the first step towards the design of future Advanced Driver Assistance Systems (ADAS) that should help the driver in increasingly complex driving tasks. Today's commercially available ACC systems are based on a single range sensor (either a radar or a laser sensor), and their use is pretty much limited to motorways or urban expressways without crossings. The traffic situations encountered are rather simple and attention can be focused on a few, well defined detected objects (cars and trucks). Nonetheless, even in these relatively simple situations, these systems show a number of limitations: they are not very good at handling fixed obstacles and may generate false alarms. Also, in some 'cut-in' situations, *ie* when the insertion of an other vehicle in the detection beam is too close to the vehicle, they may be taken by surprise.

For these systems to be more widely used, it is necessary to extend their range of operation to more complex situations in dense traffic environments around or inside urban areas. There, traffic is characterized by lower speeds, tight curves, traffic signs, crossings and "fragile" traffic participants such as motorbikes, bicycles or pedestrians. Traffic situations become very complex and it is more difficult to reliably operate an ADAS. This is mostly due to the fact that currently available sensor systems for monitoring the driving environment provide only a small part of the information required for higher level driving tasks. The way to solve this problem is to improve existing sensors like radar, laser and image processing as well as to fuse the information of these different sensor systems with appropriate scene models in order to achieve better accuracy, redundancy, robustness, and an increase of the information content.

Carsense is a European project (<http://www.carsense.org>) whose purpose is to develop a new sensing system for ADAS. It will combine several types of sensors (video, laser and radar). The focus of Carsense is on: (a) the improvement of the existing sensors, (b) the design of an on-board multi-sensor architecture, and (c) the fusion of the sensors' output.

The Sharp group at Inria Rhône-Alpes contributes to Carsense on the fusion aspects. Our goal is to demonstrate the interest of using Bayesian techniques, *ie* based on probabilistic reasoning, to address multi-sensor data fusion problems such as the Carsense one. In recent years, the probabilistic framework has become a key paradigm in Robotics. Probabilistic approaches have been used to address a wide array of robotic problems, such as CAD modeling, map building, localization, planning [1], [2], [3], [4]. The approach we intend to use is a general one, it is based on an implementation of the Bayesian theory [5]. This novel approach called *Bayesian Programming* was introduced first to design robot control programs [3], but its scope of application is much broader and it can be used whenever one has to deal with problems involving uncertain or incomplete knowledge.

The paper is organized as follows: §II overviews multi-sensor data fusion while §III presents Bayesian Programming (BP) in general. The section §IV introduces the Bayes filters, a classical approach to perform target tracking, using BP. Then the section §V presents the application of Bayes filters and BP to Carsense related problems.

II. Multi-Sensor Data Fusion

In principle, fusion of multi-sensor data provides significant advantages over single source data. In addition to the statistical advantage gained by combining same-source data (obtaining an improved estimate of a physical phenomena via redundant observations), the use of multiple types of sensors may increase the accuracy with which a phenomenon can be observed and characterized. Applications for multi-sensor data fusion are widespread, both in military and civilian areas. Ref. [6] provides an overview of multi-sensor data fusion technology and its applications.

The fusion problem addressed in Carsense is basically a *Target Tracking* problem. The objective is to collect *observations*, *ie* data from multiple sensors, on one or more potential *targets* of interest and then to estimate target characteristics such as position, velocity, etc. Because of the presence of several targets of interest in the environment, the Carsense problem falls into the *Multiple-Target Tracking* category [7].

Our primary concern within Carsense is to estimate the targets' position and velocity. It is a classical statistical estimation problem. Modern techniques involve the use of sequential estimation techniques such as the Kalman Filter or its variants. Numerous mathematical methods exist to perform coordinate transformation, observation-to-observation or observation-to-track association [8], [9], [10]. A complete and state-of-the-art review of the tracking methods with one or more sensors can be found in [7]. Challenges in this area involve situations with a large number of rapidly maneuvering targets, which is precisely the case in the traffic scenarios considered in Carsense.

III. Bayesian Programming

Any model of a real phenomenon is inherently incomplete. There are always some hidden variables, not taken into account in the model that influence the phenomenon. The effect of these hidden variables is that the model and the phenomenon never behave exactly the same way. Furthermore, perception and control are inherently uncertain. Uncertainty arises from sensor limitation or noise. Rational reasoning with incomplete and uncertain information is quite a challenge. Bayesian Programming addresses this challenge relying upon a well established formal theory: the probability theory [5].

The usual notion of *logical proposition* (either true or false) is the first key concept of probabilistic reasoning. Logical operators can be used to derive new propositions (conjunction, disjunction, negation). *Discrete variable* is the second concept that is needed: it is a set of logical proposition that are exhaustive and mutually exclusive (at least one is true, only one is true). Discrete variables can be combined too (conjunction). To deal with uncertainty, *probabilities* are attached to propositions, and to manipulate probabilities, usual inference rules are used:

- Conjunction rule: $P(X\ Y) = P(X)P(Y \mid X)$
- Normalization rule: $\sum_X P(X) = 1$

with X and Y discrete variables and P a probability.

In this framework, a Bayesian Program is made up of two parts: a *description* and a *question*.

The description can be viewed as a knowledge base containing the a priori information available on the problem at hand. It is essentially a joint probability distribution. The description is made up of three components: 1) A set of *relevant variables* on which the joint distribution is defined. Typically, variables are motor, sensory or internal. 2) A *decomposition* of the joint distribution as a product of simpler terms. It is obtained by applying Bayesian rules and taking advantages of the conditional independencies that may exists between variables. 3) The *parametric forms* assigned to each of the terms appearing in the decomposition (they are required to compute the joint distribution).

Given a distribution, it is possible to ask *questions*. Questions are obtained first by partitioning the set of variables into three sets: (1) S: the searched variables, (2) K: the known variables, and (3) F: the free variables. A question is then defined as the distribution:

$$P(S \mid K) \tag{1}$$

Given the description, it is always possible to answer a question, *ie* to compute the probability distribution $P(S \mid K)$. To do so, the following general inference is used:

$$\begin{aligned} P(S \mid K) &= \frac{\sum_F P(S\ F\ K)}{P(K)} \\ &= \frac{1}{Z} \times \sum_F P(S\ F\ K) \end{aligned} \tag{2}$$

where Z is a normalization term.

As such, the inference is computationally expensive (Bayesian inference in general has been shown to be NP-Hard [11]). A symbolic simplification phase can reduce drastically the number of sums necessary to compute a given distribution. However the decomposition of the preliminary knowledge, which express the conditional independencies of variables, still plays a crucial role in keeping the computation tractable.

We are currently developing an Application Programming Interface, which is very close to mathematical language, in order to express Bayesian programs. An inference engine has been implemented to automate Bayesian inference [3]. It operates in two stages: a) a symbolic simplification stage that permits to reduce the complexity of the probability distribution to be computed, and b) a numeric stage that actually computes the distribution.

IV. Bayes Filter

Bayes filters [12] address the general problem of estimating the state sequence x_k, $k \in \mathbb{N}$ of a target given by:

$$x_k = f_k(x_{k-1}, u_{k-1}, w_k), \tag{3}$$

where f_k is a a possibly nonlinear transition function, u_{k-1} is a "control" variable (*eg* speed or acceleration) for the sensor which allows to estimate its ego-movement between time $k-1$ and time k, and w_k is the process noise. This equation describes a Markov process of order one. The objective of tracking is to recursively estimate x_k from the sensor measurements:

$$z_k = h_k(x_k, v_k). \tag{4}$$

where h_k is a possibly nonlinear function and v_k is the measurement noise. This function models the uncertainty of the measurement z_k of the state x_k of a detected target.

The goal of the tracking is to recursively estimate the probability distribution $P(X_k \mid Z_k)$, known as the *posterior* distribution. In general, this estimation is done in two stages: *prediction* and *estimation*. The goal of prediction is to compute an *a priori* estimate of the target's state known as the *prior* distribution. The goal of estimation is to compute the *posterior* distribution, using this a priori and the measurement of the sensor.

Exact solutions to this recursive propagation of the posterior density do exist in a restrictive set of cases. In particular, the Kalman filter [13], [14] is an optimal solution when the functions f_k and h_k are linear and the noises w_k and v_k are Gaussian. But in general, solutions cannot be determined analytically.

In the following, the two stages, prediction and estimation, of Bayes filters will be presented using the formalism introduced in §III.

A. Prediction stage.

The goal of this stage is to estimate an *a priori* on the target's state distribution using the latest estimation of this state, and the Markov model (3). The variables that are relevant here are:

- X_k: the state of the target at time k;
- X_{k-1}: the state of the target at time $k-1$;
- U_{k-1}: the "control" input of the sensor at time $k-1$. For example, it could be a measurement of the instantaneous velocity at time $k-1$.

The following decomposition of the joint distribution determined by these three variables can be obtained:

$$P(X_k X_{k-1} U_{k-1}) = P(X_{k-1})P(U_{k-1})P(X_k|X_{k-1}U_{k-1}). \tag{5}$$

Parametric forms must be assigned to each of the terms appearing in the decomposition:

- $P(X_{k-1})$ represents the information on the target's state at time $k-1$. When available, the *posterior* distribution at time $k-1$ is used to specify $P(X_{k-1})$.
- $P(U_{k-1})$ represents the information on the control of the sensor.
- $P(X_k|X_{k-1}\ U_{k-1})$ is specified by (3).

Now the description is complete and question can be asked. Since we want an estimate of the target's state, we ask the inference engine to answer the question:

$$P(X_k \mid U_{k-1}). \tag{6}$$

Following the general inference scheme given by (2), the result of the inference is:

$$\begin{aligned} P(X_k|U_{k-1}) &= \sum_{X_{k-1}} P(X_k\ X_{k-1}\ U_{k-1}) \\ &= P(U_{k-1}) \sum_{X_{k-1}} P(X_{k-1})P(X_k|X_{k-1}\ U_{k-1}). \end{aligned}$$

Due to the summation on the variable X_{k-1}, the uncertainties on the target's state at time $k-1$ are propagated to the prior distribution.

B. Estimation Stage

The goal of this stage is to estimate the *posterior* distribution on the target's state, using the current sensor observation. The variables relevant here are:

- X_k: the target's state at time k.
- Z_k: the sensor observation at time k.

We can write the following decomposition of the joint distribution determined by this two variables:

$$P(X_k\ Z_k) = P(X_k)P(Z_k \mid X_k). \tag{7}$$

Parametric forms must be assigned to each of the terms appearing in the decomposition:

- $P(X_k)$ represents the *a priori* information on the target's state at time k. Thus the *prior* distribution is used to specify $P(X_k)$. If this prior distribution is not available (for the first estimation for example), a uniform distribution could be selected.
- $P(Z_k \mid X_k)$ represents the response of the sensor to a target located at X_k, given by (4).

Now the description is complete and question can be asked. As we want an estimation of the target's state, we ask the inference engine to answer the question:

$$P(X_k|Z_k). \tag{8}$$

This formulation of Bayes filters is valid for one target and one sensor. The next section will extend this formulation so as to perform tracking of multiple targets using multiple sensors.

V. Application to Carsense

The problem addressed in Carsense is basically a *Multiple Sensor Multiple target Tracking* problem: given a set of sensor information on the driving environment, our concern is to determine the actual set of targets of interest and to estimate some of their characteristics, mainly position and velocity.

In this section, we model the multiple target tracking problem as a general Bayes filter problem. During the estimation stage, an *observation-to-observation* association is required. The goal of this association is to partition the set of observations provided by the different sensors into set of observations produced by the same source target. In [15], results on this association were presented. In particular, it has been shown that this association could be done simultaneously with the estimation step. The posterior distribution is then multi-modal, each peak corresponding to a target in the environment.

In [15], the assumption was made that the different sensors were synchronous. This hypothesis can't be made in the Carsense frameworks since all sensors works asynchronously. In highly dynamic environments, such as urban or suburban roads, this timing error can't be neglected.

In this section, we extend our association-estimation mechanism to the asynchronous case. The time difference between the sensor measurements and the target's state estimation are taken into account directly in the sensor model.

A. Sensor Modeling

Sensor model such as the one we have introduced in [15] is valid if the measurement Z_k and the target's state X_k correspond to the same instant k. Here we consider a sensor measurement Z_{t_1} and the target's state X_{t_0}, with $t_0 > t_1$. $t_0 - t_1$ is noted Δt. We introduce a new variable X_{t_1}, the target's state a time t_1. The "control" of the sensor at time t_0, U_{t_0} is also considered.

The following decomposition of the joint distribution determined by these variables can be obtained:

$$P(Z_{t_1}\, X_{t_1}\, X_{t_0}\, \Delta t\, U_{t_0}) = P(\Delta t)P(U_{t_0})P(X_{t_0}) \\ P(X_{t_1}|X_{t_0}\, \Delta t\, U_{t_0})P(Z_{t_1}|X_{t_1})$$

Parametric forms must be assigned to each of the terms appearing in the decomposition:

- $P(\Delta t)$, $P(U_{t_0})$, $P(X_{t_0})$ could be chosen as uniform distributions.
- $P(X_{t_1} \mid X_{t_0}\, \Delta t\, U_{t_0})$ is set according to a dynamic model including the target movement and the ego-movement of the sensor. Note that this model is not exactly the same as the one defined by (3), because here $t_0 < t_1$.
- $P(Z_{t_1}|\, X_{t_1})$ is the sensor model defined in [15], including the probability detection.

Now the description is complete and questions can be asked. Since we want to estimate the sensor response, we ask the inference engine:

$$P(Z_{t_1} \mid X_{t_0}\, \Delta t\, U_{t_0}) \tag{9}$$

The estimation stage has to be re-written to take into account this new sensor model.

B. Estimation Stage

We want to estimate the posterior distribution at time t_0, using the available sensors observations. We consider the case where we have a set of S sensors S_i, each returning O^i observations.

The variables relevant here are:

- X_{t_0}: the state of the target of interest at time t_0.
- Z_i^j: the j^{th} observation of the sensor S_i. To simplify the notation, the timestamp is not precised.
- Δt_i^j: the difference of time between Z_i^j and t_0.
- U_{t_0}: the "control" of the sensor at time t_0.
- M_i: it is a matching variable that indicates wich observation of the sensor S_i corresponds to the target of interest. A special case is added to the O^i values that M_i can take, which means that no observation of the sensor S_i correspond to the target of interest.

Before proceeding to the decomposition of the joint distribution determined by these variables, a few reasonable assumptions are made:

- The sensors' observations are conditionally independent *wrt* the target's position and the corresponding matching variable:

$$P(Z_i^j|Z_n^l X_{t_0} \Delta t_i^j\, U_{t_0} M_i M_n) = P(Z_i^j|X_{t_0} \Delta t_i^j\, U_{t_0} M_i);$$

- The matching variables $M_i, i = 1 \ldots S$ are conditionally independent *wrt* the target's position:

$$P(M_i \mid X_{t_0}\; M_n) = P(M_i \mid X_{t_0});$$

- The observations of a given sensor S_i are conditionally independent *wrt* the target's position.

$$P(Z_i^j \mid X_{t_0}\; \Delta t_i^j\; U_{t_0}\; Z_i\; M_i) = P(Z_i^j \mid X_{t_0}\; \Delta t_i^j\; M_i).$$

Thanks to these assumptions and using the Bayesian inference rules, we can write the following decomposition of the joint distribution:

$$P(X_{t_0}\, Z_1^1 \ldots Z_S^{O^S}\, M_1 \ldots M_S\, \Delta t_1^1 \ldots \Delta t_S^{O^S}\; U_{t_0}) = \\ P(X_{t_0})P(U_{t_0}) \prod_{i=1}^{S} \left(\frac{P(M_i|X_{t_0})}{\prod_{j=1}^{O^i} P(\Delta t_i^j)P(Z_i^j|X_{t_0}\, U_{t_0}\, \Delta t_i^j\, M_i)} \right)$$

Finally, parametric forms are assigned to each of the terms appearing in the decomposition:

- $P(X_{t_0})$ represents the a priori information on the different targets' positions. Thus the prior distribution is used to specify it.
- $P(U_{t_0})$ represents the information on the control of the sensor.
- $P(M_i \mid X_{t_0})$ is uniform since knowing the target's state only does not suffice to determine M_i.
- The form of $P(Z_i^j | X_{t_0} U_{t_0} \Delta t_i^j M_i)$ depends on the value of M_i:
  - If $M_i = j$ then Z_i^j is an observation of the target of interest by the sensor S_i, and the form of $P(Z_{k,i}^j | X_{t_0} U_{t_0} \Delta t_i^j M_i)$ is the sensor model defined by (9).
  - If $M_i \neq j$, Z_i^j is not an observation of the target of interest. The distribution is uniform.

Now the description is complete and questions can be asked. To evaluate the posterior distribution, we just have to ask the inference engine to solve the question

$$P(X_{t_0} \mid Z_1^1 \ \ldots \ Z_S^{O^S} \ \Delta t_1^1 \ \ldots \ \Delta t_S^{O^S} \ U_{t_0}) \qquad (10)$$

C. Experimental Results

To test the complete algorithm of tracking, we have used the platform of simulation of data fusion and perception that was developed at the LIVIC[1] [16]. This platform can simulate a car and its environment. In our experiments, the car was equipped with two sensors working asynchronously and with different fields of view (thus a target could be seen by only one sensor).

Fig. 1 shows a short sequence of a complete tracking involving and four targets. The state of a target is given by its position (distance ρ and heading θ). The middle column depicts a top view of the situation. The left column shows the *prior* distribution. Finally the right column shows the *posterior* distribution, *ie* the answer to the question:

$$P(\rho_t \theta_t \mid \rho_1^i \theta_1^i \Delta t_1^i \rho_2^j \theta_2^j \Delta t_2^j U_t), \ \begin{matrix} i = 1 \ldots O^{S_1}, \\ j = 1 \ldots O^{S_2}. \end{matrix} \qquad (11)$$

We suppose that at time $t = 1.6s$, no prediction step is available. Three targets have been detected by the two sensors: two by the sensor S_1, and one by the sensor S_2. Then the posterior distribution at $t = 1.6s$ presents three modes. These three modes are present in the *prior* distribution at $t = 1.7s$. At this time, the second sensor detects a fourth target. So a fourth mode appears in the *posterior* distribution. But its probability is smaller than the one associated to the previously detected targets. At $t = 1.9s$, the first sensor have lost one target. Due to the prediction step, a mode still appears in the posterior distribution, but its probability has fallen. As this target is still not detected at $t = 2.0$, the mode disappear in the *posterior* distribution.

[1] http://www.inrets.fr/ur/livic/

This short sequence illustrates how new object's appearing and disappearing are treated by the Bayes filter. Nevertheless, precautions have to be taken, because the optimization algorithms requested for the computation of the distributions could have effects on these distributions.

VI. Conclusion

This paper has addressed the problem of multi-sensor data fusion with a new programming technique based on Bayesian inference. This method, called *Bayesian Programming*, has been illustrated in an automotive application. Several examples were given in order to emphasize the main advantages of the approach, namely: a) a clear and well-defined mathematical background. b) a generic and uniform way to formulate problems. c) the importance of explicitly modeling key problem features (such as the sensors' performance in our examples). Future developments will include: *a) experiments with real sensor data:* so far, the experiments have been conducted with simulated sensor data. The sensors mounted on the Carsense test-bed vehicle will be used (laser, radar and video sensors.) *b) higher-level sensor fusion:* we would like to take advantage of the generality and the expressive power of Bayesian Programming in order to go beyond the 'simple' estimation of the targets' position and velocity, to take into account additional observations, *eg* the shape/type of a target or the lane within which it is moving, so as to perform *target classification* in terms of "dangerous or not", and if possible, in terms of "cars, trucks, cycles, pedestrians, etc."

Acknowledgements. This work was partially supported by the European project IST-1999-12224 *"Sensing of Car Environment at Low Speed Driving"* (`http://www.carsense.org`).

VII. References

[1] S. Thrun, W. Burgard, and D. Fox. A probalistic approach to concurrent mapping and localization for mobile robots. *Machine Learning and Autonomous Robots*, 31(5), 1998.

[2] L.P. Kaelbling, M.L. Littman, and A.R. Cassandra. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101, 1998.

[3] O. Lebeltel, P. Bessière, J. Diard, and E. Mazer. Bayesian robots programming. Research Report 1, Les Cahiers du Laboratoire Leibniz, Grenoble (FR), May 2000.

[4] K Mekhnacha, E Mazer, and P Bessière. The design and implementation of a bayesian CAD modeler for robotic applications. *Advanced Robotics*, 15(1):45–70, 2001.

[5] E. T. Jaynes. Probability theory: the logic of science. in press, see `http://bayes.wustl.edu/`, 1995.

[6] D. Hall and J. Llinas. An introduction to multisensor data fusion. *IEEE Proceedings*, 85(1), January 1997.

[7] S. Blackman and R. Popoli. *Design and Analysis of Modern Tracking Systems*. Artech House, 2000.

[8] Y. Bar-Shalom and X. Li. *Multitarget-Multisensor Tracking: Principles and Techniques*. Artech House, 1995.

[9] H. Gauvrit, J.P. Le Cadre, and C Jauffret. A formulation of multitarget tracking as an incomplete data problem. *IEEE Trans. on Aerospace and Electronic Systems*, 33(4), 1997.

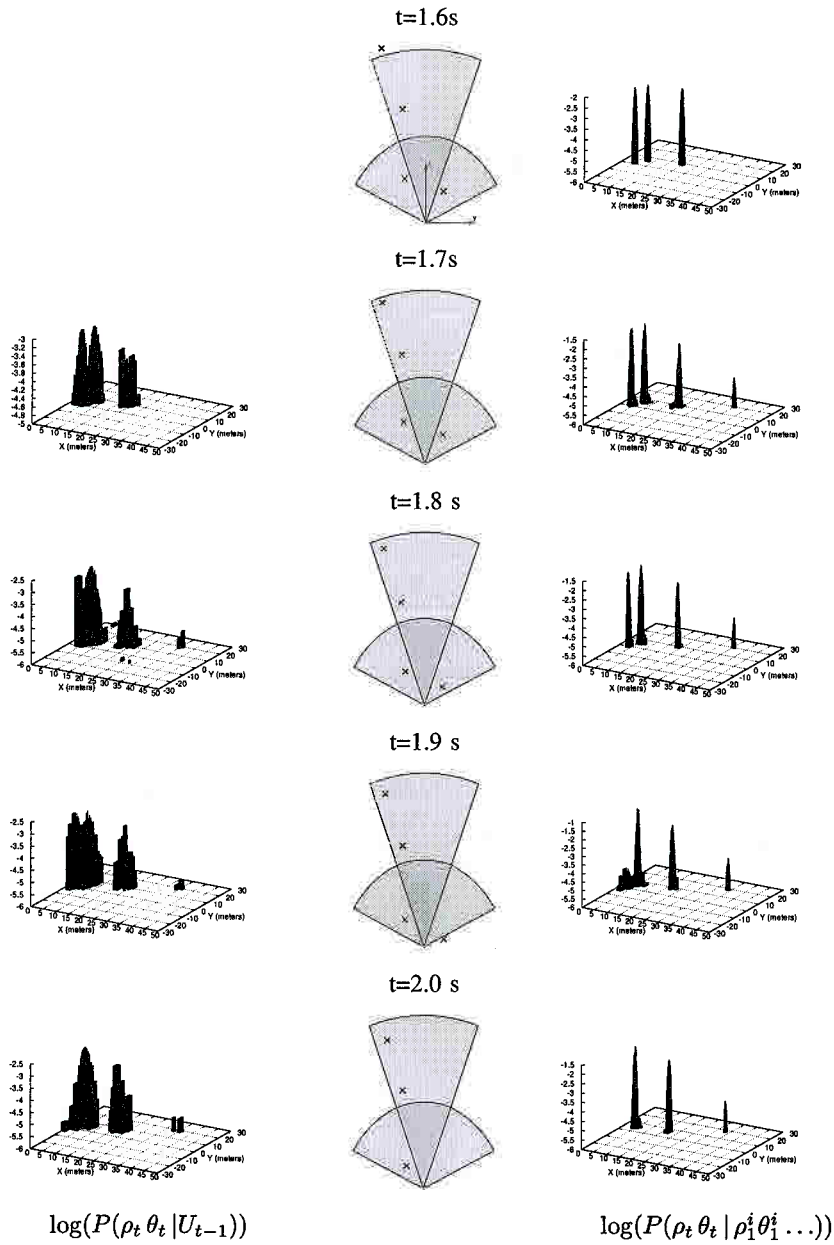


Fig. 1. a short sequence of a complete tracking. Vertical axis is the logarithm of the probability distributions

[10] R.L. Streit and T.E. Luginbuhl. Probabilistic multi-hypothesis tracking. Technical Report 10,428, Naval Undersea Warfare Center Division Newport, 1995.

[11] G. Cooper. The computational complexity of probabilistic inference using bayesian belief network. *Artificial Intelligence*, 42(2-3), 1990.

[12] L.D. Stone, C.A. Barlow, and T.L Corwin. *Bayesian Multiple Target Tracking*. Artech House, 1999.

[13] R.E. Kalman. A new approach to linear filtering and predictions problems. *Journal of Basic Engineering*, March 1960.

[14] G. Welch and G. Bishop. An introduction to the kalman filter. `http://www.cs.unc.edu/~welch/`.

[15] C. Coué, Th. Fraichard, P. Bessière, and E. Mazer. Multi-sensor data fusion using bayesian programming: an automotive application. In *Proc. of the IEEE-RSJ Int. Conf. on Intelligent Robots and Systems*, Lausanne (CH), September-October 2002.

[16] D. Gruyer, M. Mangeas, and R. Alix. Multi-sensor fusion approach for driver assistance systems. In *IEEE ROMAN*, Bordeaux, France, 2001.