Multi-Sensor Data Fusion Using Bayesian Programming: an Automotive Application

Christophe Coué, Thierry Fraichard, Pierre Bessiere, Emmanuel Mazer

Multi-Sensor Data Fusion Using Bayesian Programming : an Automotive Application

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, *i.e.* 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 characterised 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[1] 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, *i.e.* 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 modelling, map building, localisation, 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 organised as follows: §II overviews multi-sensor data fusion while §III presents Bayesian Programming (BP) in general. The next two sections focus on the Carsense context: first, §IV describes how to model the sensors in the Bayesian framework. Then §V presents the application of BP to different 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 characterised. Applications for multi-sensor data fusion are widespread, both in

[1]Project IST-1999-12224 *"Sensing of Car Environment at Low Speed Driving"* <http://www.carsense.org>.

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*, *i.e.* data from multiple sensors, on one or more potential *targets* of interest and then to partition the observations into *tracks*, *i.e.* sets of observations produced by the same target. Once this *association* is done, *estimation* can take place: target characteristics such as position, velocity, etc. are computed for each track. 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 manoeuvring 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:
$$P(X\ Y) = P(X)P(Y \mid X) = P(Y)P(X \mid Y)$$
- Normalisation rule:
$$\sum_X P(X) = 1$$

whit 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:

- A set of *relevant variables* on which the joint distribution is defined. Typically, variables are motor, sensory or internal.
- 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.
- 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) *Searched*: the searched variables, (2) *Known*: the known variables, and (3) *Free*: the free variables. A question is then defined as the distribution:

$$P(Searched \mid Known) \tag{1}$$

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

$$\begin{aligned} & P(Searched \mid Known) \\ &= \sum_{Free} P(Searched\ Free \mid Known) \\ &= \frac{\sum_{Free} P(Searched\ Free\ Known)}{P(Known)} \\ &= \frac{1}{Z} \times \sum_{Free} P(Searched\ Free\ Known) \end{aligned}$$

where Z is a normalisation 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 API[2], 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. Sensor Modelling

A. Sensor Models

Be it for association or estimation purposes, it is fundamental to have a model of the sensor's performance, of the reliability and precision of the observations obtained. To address this modelling issue, we have defined stochastic

[2]Application Programming Interface

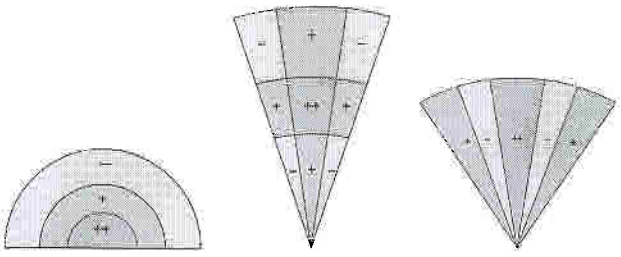

Fig. 1. Sensors' precision *w.r.t.* the target's position in the field of view.

sensor models that are general enough to represent most existing sensors.

We present now the general model we have designed for a range sensor measuring the heading and direction of a target. It takes into account the target detection probability, the sensor noise that corrupt the range measurement, and the variation of range precision *w.r.t.* the target's position.

The uncertainty in the measurement of the distance ρ and the heading θ of a detected target whose actual distance and heading is (z, α), is modelled by the probability distribution $P(\rho\ \theta|det(z,\alpha))$ where $det(z,\alpha)$ is a boolean variable indicating if a detection of a target occurred at position (z,α). To simplify the model, we consider that the measures in distance and heading are independent[3] so that the probability distribution can be written as:

$$P(\rho\ \theta|det(z,\alpha)) = P(\rho|det(z,\alpha))P(\theta|det(z,\alpha))$$

with:

$$\begin{aligned} P(\rho|det(z,\alpha)) &= G_\rho(z,\sigma_\rho(z,\alpha)) \\ P(\theta|det(z,\alpha)) &= G_\theta(\alpha,\sigma_\theta(z,\alpha)), \end{aligned}$$

G_ρ (resp. G_θ) is a Gaussian distribution on the variable ρ (resp. θ) whose mean value is z (resp. α) and whose standard deviation is $\sigma_\rho(z,\alpha)$ (resp. $\sigma_\theta(z,\alpha)$). Note that both σ_ρ and σ_θ are functions of z and α: they model the variation of the observation's precision (in distance and heading) *w.r.t.* the target's position.

To model the sensor reliability, *i.e.* its ability to detect a target, we introduce a target detection probability $P_d(z,\alpha)$ which is a function of the target's position. Now we have:

$$P(det(z,\alpha)|z\ \alpha) = P_d(z,\alpha)$$

and the missed detection probability is of course given by:

$$P(\neg det(z,\alpha)|z\ \alpha) = 1 - P_d(z,\alpha)$$

Finally, the response of the sensor to a given target is given by:

$$\begin{aligned} P(\rho\ \theta|z\ \alpha) &= P(\rho\ \theta|det(z,\alpha))P_d(z,\alpha) \\ &+ P(\rho\ \theta|\neg det(z,\alpha))(1 - P_d(z,\alpha)) \end{aligned} \quad (2)$$

This class of stochastic sensor models can be applied to a large variety of sensors including radars and lasers. The parameters required to model an actual sensor, *i.e.* $P_d(z,\alpha)$, $\sigma_\rho(z,\alpha)$ and $\sigma_\theta(z,\alpha)$, should be determined either thanks to the sensor manufacturer's specifications or through calibration.

[3]This assumption is not true in general, it could be lifted easily.

B. Experimental Sensors

We have used the model defined earlier to define three different types of sensors that will be used later in our experiments on association and estimation.

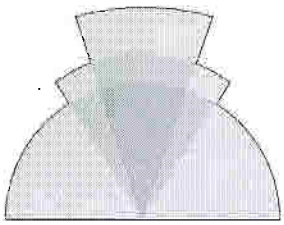

Fig. 2. Layout of the three sensors' field of views.

Fig. 1 depicts the field of views of the different sensors and the variation of the observation's precision (in distance and heading) *w.r.t.* the target's position in each field of view. "++" means that the sensor is really accurate in this area, "+" means that it is less accurate and "-" means that it is even less accurate. Outside its field of view, the sensor does not perceive anything. It is the setting of $P_d(z,\alpha)$, $\sigma_\rho(z,\alpha)$ and $\sigma_\theta(z,\alpha)$ that determines the field of view of a sensor and the different areas within.

Fig. 2 depicts the layout of the different sensors' field of views. Note that parts of the environment are observed by zero, one, two or three sensors.

V. Application to Carsense

We contribute to Carsense on the fusion aspects. Given a set of sensors providing information on the driving environment and more precisely on a set of potential targets, our primary concern is to determine the actual set of targets of interest and to estimate some of their characteristics, mainly position and velocity. As mentioned earlier, such a multiple-target tracking problem involves two steps:

- Association: regrouping of the sensors' observations into tracks associated with the same source target.
- Estimation: actual computation of the target characteristics.

To demonstrate the generality and the power of BP, we first show how it can handle these two problems separately (§V-A and §V-B). Then we show how it can solve them both simultaneously (§V-C).

A. Estimation

We consider here that the association step is done. The observations have been sorted out and associated to a potential target. We start with a simple example involving one target and three different sensors providing information on the target's position (heading and distance). To solve the estimation problem at hand using Bayesian Programming, we have first to specify the different components of

the Bayesian program, *i.e.* the variables, the decomposition and the parametric forms.

A.1 Program Specification

The variables relevant here are:

- Actual distance z and heading α of the target.
- Distance ρ_i and heading θ_i measured by the sensor $S_i, i = 1 \ldots 3$.

Altogether eight variables that determine the joint distribution:

$$P(z\ \alpha\ \rho_1\ \theta_1\ \rho_2\ \theta_2\ \rho_3\ \theta_3) \tag{3}$$

It will be assumed that the sensors' observations are conditionally independent *w.r.t.* the target's position meaning that:

$$P(\rho_i\ \theta_i | z\ \alpha\ \rho_j\ \theta_j) = P(\rho_i\ \theta_i | z\ \alpha)$$

In other words, knowing (z, α), the knowledge of (ρ_j, θ_j) does not bring extra information to (ρ_i, θ_i). Note that this is very different to saying that $P(\rho_i\ \theta_i | \rho_j\ \theta_j) = P(\rho_i\ \theta_i)$, *i.e.* that the sensors are independent. This is clearly false, when two sensors are measuring the same target, their observations are obviously dependent.

Thanks to this conditional independence assumption and using the Bayesian inference rules, we can write the following decomposition of (3):

$$\begin{aligned} &P(z\ \alpha\ \rho_1\ \theta_1\ \rho_2\ \theta_2\ \rho_3\ \theta_3) = \\ &\quad P(z\ \alpha) P(\rho_1\ \theta_1 | z\ \alpha) P(\rho_2\ \theta_2 | z\ \alpha) P(\rho_3\ \theta_3 | z\ \alpha) \end{aligned} \tag{4}$$

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

- $P(z\ \alpha)$ represents the a priori information on the target's position. If a priori information is available (from a previous tracking stage for instance), it could be used to specify $P(z\ \alpha)$. Otherwise, a uniform distribution should be selected.
- $P(\rho_i\ \theta_i | z\ \alpha)$ represents the response of the sensor S_i to a target located at (z, α). The parametric form chosen for this distribution is the sensor model we have defined in §IV

Now the description is complete and questions can be asked. To estimate the target's position given the sensors' observations, we ask the inference engine to answer the following question:

$$P(z\ \alpha | \rho_1\ \theta_1\ \rho_2\ \theta_2\ \rho_3\ \theta_3) \tag{5}$$

A.2 Experimental Results

Fig. 3 depicts a situation where the target is in the field of view of the three sensors and is detected by all of them. Figs. 3a, 3b and 3c represent the probability distribution of the target position knowing the different sensor responses. Note the difference in precision of the different sensors. Fig. 3d represents the estimation result. In this case, it is close to the observation returned by S_2, the most accurate sensor.

Fig. 4 illustrates the robustness of the approach *w.r.t.* to sensor failures. In this example, sensor S_2 fails to detect the target. Since our sensor model integrates the detection probability, it can handle this situation easily and

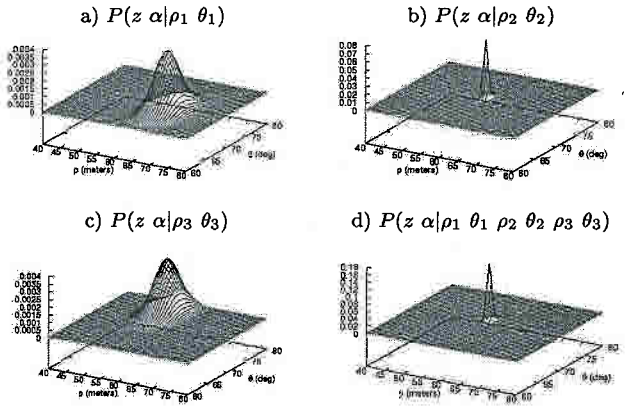

Fig. 3. Estimation example #1: the three sensors detect the target.

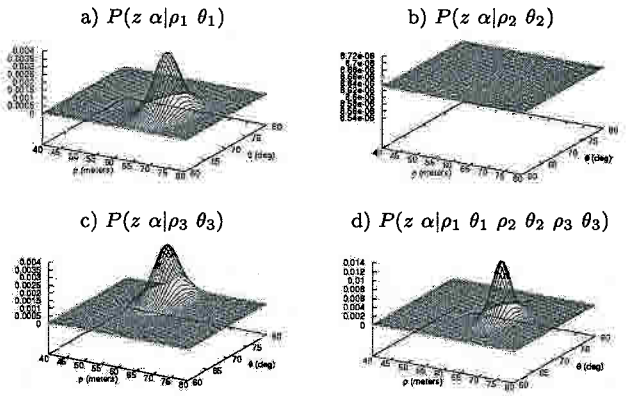

Fig. 4. Estimation example #2: sensor S_2 fails to detect the target.

$P(z\ \alpha|\rho_2\ \theta_2)$ becomes a uniform distribution. Now estimation takes place exactly as before, the only difference is that the final result is less accurate.

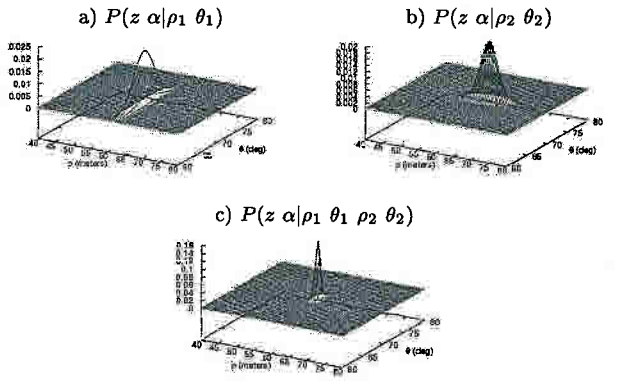

Fig. 5. Estimation example #3: importance of explicitly modelling the sensor's performance.

Fig. 5 illustrates the importance of explicitly modelling the sensor's performance. In this example, there are only two sensors: the first one is accurate in distance but not

in heading. The second one is accurate in heading but not in distance. The estimation with these two sensors yields a result which is accurate both in distance and heading.

TABLE I
STATISTICAL DATA FOR ESTIMATION.

	S_1	S_2	S_2	Estimation
Avg. Error #1	3.66	1.28	1.65	1.03
# Detections	591	203	430	600
Avg. Error #2	3.53	1.95	2.56	2.34

Besides these examples, we have carried out further experiments in order to gather statistical data. An experimental run would go like this: first, a target's position is selected randomly. Given this position, the different sensor models are used to simulate the sensor's observations. Then the estimation question (5) is solved. Finally, to estimate the position of the target from the answer to question (5), we use an optimisation algorithm to determine the maximum value of this probability distribution. This optimisation is based on a genetic algorithm [12]. The results obtained are summarised in Table I.

First, we made 150 runs with targets selected randomly inside the common field of view of the three sensors. Line "Avg. Error #1" gives the average error between the actual position of the target and the position of the target given by the different sensors and the estimation. It illustrates the interest of using several sensors with different precisions: the estimation precision is better.

Second, we made 1000 runs with targets selected randomly: they can fall inside or outside the field of view of the different sensors. Line "# Detections" indicates how many times the target has been detected by the different sensors. 600 times out of 1000, the target was detected by at least one sensor and estimation could take place. Once again, it illustrates the interest of using several sensors: the target is detected more often with three sensors than with one only (whichever one). Line "Avg. Error #2" gives the average error in this case. It may appear that the estimation yields results that are less accurate than the results obtained with the sensor S_2 only. However, do keep in mind that, among the 600 estimations made, S_2 contributed to only 203 of them.

B. Association

Association is about partitioning a set of observations provided by different sensors into tracks, *i.e.* sets of observations produced by the same source target. We consider the case where we have a set of S sensors $S_i, i = 1 \dots S$, each returning O^i observations (heading and distance).

B.1 Program Specification

The variables relevant here are:

- (z, α): distance and heading of the target of interest (although there may be several targets, only one (z, α) is needed).
- (ρ_i^j, θ_i^j): the j^{th} observation made by the sensor S_i.
- M_i: it is a matching variable that indicates which observation of the sensor S_i corresponds to the target of interest.

Altogether $2 + 2\sum_i O^i + S$ variables that determine the joint distribution:

$$P(z\ \alpha\ \rho_1^1\ \theta_1^1\ \rho_1^2\ \theta_1^2 \dots \rho_S^{O^S}\ \theta_S^{O^S}\ M_1 \dots M_S) \tag{6}$$

Before proceeding to the decomposition of (6), a few reasonable assumptions are made:

- The sensors' observations are conditionally independent *w.r.t.* to the target's position and the corresponding matching variable (*cf.* §V-A).

$$P(\rho_i^j\ \theta_i^j | z\ \alpha\ \rho_k^l\ \theta_k^l\ M_i\ M_k) = P(\rho_i^j\ \theta_i^j | z\ \alpha\ M_i)$$

- The matching variables $M_i, i = 1 \dots S$ are conditionally independent *w.r.t.* to the target's position.

$$P(M_i | z\ \alpha\ M_j) = P(M_i | z\ \alpha)$$

- The observations of a given sensor S_i are conditionally independent *w.r.t.* to the target's position and the corresponding matching variable.

$$P(\rho_i^j\ \theta_i^j | z\ \alpha\ \rho_i^k\ \theta_i^k\ M_i) = P(\rho_i^j\ \theta_i^j | z\ \alpha\ M_i)$$

Once again, thanks to these assumptions and using the Bayesian inference rules, we can write the following decomposition of (6):

$$P(z\ \alpha\ \rho_1^1\ \theta_1^1\ \rho_1^2\ \theta_1^2 \dots \rho_S^{O^S}\ \theta_S^{O^S}\ M_1 \dots M_S) = P(z\ \alpha) \prod_{i=1}^{S} \left(P(M_i | z\ \alpha) \prod_{j=1}^{O^i} P(\rho_i^j\ \theta_i^j | z\ \alpha\ M_i) \right) \tag{7}$$

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

- $P(z\ \alpha)$ represents the a priori information on the target's position.
- $P(M_i | z\ \alpha)$ is uniform since knowing the target's position only does not suffice to determine M_i.
- The form of $P(\rho_i^j\ \theta_i^j | z\ \alpha\ M_i)$ depends on the value of M_i:
  - If $M_i = j$ then $(\rho_i^j\ \theta_i^j)$ is an observation of the target of interest by the sensor S_i, and the form of $P(\rho_i^j\ \theta_i^j | z\ \alpha\ M_i)$ is the sensor model defined in §IV.
  - If $M_i \neq j$, $(\rho_i^j\ \theta_i^j)$ is not an observation of the target of interest. The distribution is uniform.

Now the description is complete and questions can be asked. To solve the association problem, we ask the inference engine to answer the following question:

$$P(M_1 \dots M_S | \rho_1^1\ \theta_1^1\ \rho_1^2\ \theta_1^2 \dots \rho_S^{O^S}\ \theta_S^{O^S}) \tag{8}$$

B.2 Experimental Results

To test the association, we have defined the following experiment: two sensors are used to detect two targets placed right in front of the vehicle at a distance of 50 m. with a 3 m. interval between them. In this case, $S = O^1 = O^2 = 2$, the question asked is:

$$P(M_1\ M_2 | \rho_1^1\ \theta_1^1\ \rho_1^2\ \theta_1^2\ \rho_2^1\ \theta_2^1\ \rho_2^2\ \theta_2^2) \tag{9}$$

TABLE II

ASSOCIATION : SENSITIVITY TO THE SENSORS' PRECISIONS.

$\sigma_\theta(z,\alpha) =$	0.5	2	5
$M_1 = 1, M_2 = 1$	0.294	0.364	0.239
$M_1 = 2, M_2 = 1$	$6e^{-5}$	0.031	0.273
$M_1 = 1, M_2 = 2$	$6e^{-5}$	$4e^{-3}$	0.0931
$M_1 = 2, M_2 = 2$	0.704	0.594	0.386

and it is possible to compute the probability of the four possible associations: $(M_1 = 1, M_2 = 1)$ or $(M_1 = 2, M_2 = 1)$ or $(M_1 = 1, M_2 = 2)$ or $(M_1 = 2, M_2 = 2)$. Note that $P(M_1 = i, M_2 = j)$ must be interpreted as the probability that the i^{th} observation of S_1 is associated with the j^{th} observation of S_2.

We carried several experimental runs in order to show the influence of the sensors' precisions in the association and therefore the importance of explicitly modelling the sensors' performance. Table II illustrates the outcome of these experiments: it contains the probability of each possible association for decreasing precisions in θ (for both sensors). From this table, it can be seen that, with accurate sensors (column 1), the association is reliable: two association probabilities stand out. When the precision decreases, it becomes less and less obvious.

C. Association and Estimation

C.1 Program Specification

In the Bayesian framework, it turns out that solving the association and the estimation problems simultaneously can be achieved with the description defined in V-B for the association problem. We keep the same description and simply ask a new question:

$$P(z\ \alpha|\rho_1^1\ \theta_1^1\ \rho_1^2\ \theta_1^2 \ldots \rho_S^{O^S}\ \theta_S^{O^S}) \tag{10}$$

C.2 Experimental Results

To test the simultaneous association and estimation, we have used the same experiment as in §V-B and asked the question:

$$P(z\ \alpha|\rho_1^1\ \theta_1^1\ \rho_1^2\ \theta_1^2\ \rho_2^1\ \theta_2^1\ \rho_2^2\ \theta_2^2) \tag{11}$$

The output of the inference is a two-dimensional probability distribution which is multi-modal: it should contains as many peaks as there are targets in the environment. Fig. 6 depicts such a multi-modal distribution. Now, given this distribution, it suffices to search all its modes in order to identify the different targets.

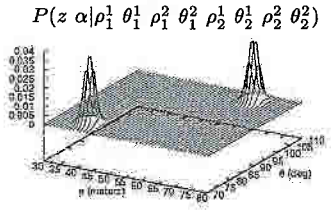

Fig. 6. Simultaneous association and estimation example.

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 emphasise 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 modelling 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 synthetic sensor data obtained thanks to virtual sensors. The next step will be to use models real sensors models and data. The sensors mounted on the Carsense testbed vehicle will be used. The vehicle is equipped with 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, *e.g.* 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 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`).

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 (Joint Issue)*, 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. Unprinted book, available at `http://bayes.wustl.edu/`, 1970's.

[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.* CT:YBS Publishing, 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), October 1997.

[10] R.L. Streit and T.E. Luginbuhl. Probabilistic multi-hypothesis tracking. Technical Report 10,428, NUWC Newport, 1995.

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

[12] D. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning.* Addison-Wesley, 1989.