Online ISSN 1848-3380, Print ISSN 0005-1144
ATKAFF 55(4), 386–398(2014)

Ivan Marković, Ivan Petrović

Bayesian Sensor Fusion Methods for Dynamic Object Tracking — A Comparative Study

DOI 10.7305/automatika.2014.09.847
UDK 681.586.015:004.62.052
IFAC 3.0.2; 4.2

Original scientific paper

In this paper we study the problem of Bayesian sensor fusion for dynamic object tracking. The prospects of utilizing measurements from several sensors to infer about a system state are manyfold and they range from increased estimate accuracy to more reliable and robust estimates. Sensor measurements may be combined, or fused, at a variety of levels; from the raw data level to a state vector level, or at the decision level. In this paper we mainly focus on the Bayesian fusion at the likelihood and state vector level. We analyze two groups of data fusion methods: centralized independent likelihood fusion, where the sensors report only its measurement to the fusion center, and hierarchical fusion, where each sensor runs its own local estimate which is then communicated to the fusion center along with the corresponding uncertainty. We compare the prospects of utilizing both approaches, and present explicit solutions in the forms of extended information filter, unscented information filter, and particle filter. Furthermore, we also propose a solution for fusion of arbitrary filters and test it on a hierarchical fusion example of two of the aforementioned filters. Hence, the main contributions of this paper are systematic comparative study of Bayesian fusion methods, and a method for hierarchical fusion of arbitrary filters. The fusion methods are tested on a synthetic data generated by multiple Monte Carlo runs for tracking of a dynamic object with several sensors of different accuracies by analyzing the quadratic Rényi entropy and root-mean-square error.

Key words: Bayesian sensor fusion, Information filter, Particle filter, Rényi entropy

Komparativna studija Bayesovih metoda fuzije u svrhu praćenja gibajućih objekata. U ovom članku razmatra se problem Bayesove fuzije senzora u svrhu praćenja gibajućih objekata. Prednosti korištenja mjerenja više senzora kako bi se estimiralo stanje sustava su višestruke te se kreću od povećanja preciznosti do pouzdanijih i robusnijih estimacija. Fuzija mjerenja senzora može se izvršiti na razini neobrađenih mjerenja, razini estimacije stanja te na još višoj razini–razini donošenja odluka. U ovom se članku fokusira na Bayesovu fuziju senzora na razini funkcija vjerodostojnosti i na razini vektora stanja sustava. Analiziramo dvije grupe metoda fuzije podataka: centraliziranu fuziju nezavisnih funkcija vjerodostojnosti, u kojoj senzori šalju samo svoja mjerenja centru fuzije, i hijerarhijsku fuziju, gdje svaki senzor lokalno estimira stanje sustava koje se potom šalje centru fuzije zajedno sa pripadajućom nesigurnosti. Uspoređujemo prednosti korištenja oba pristupa te predstavljamo eksplicitna rješenja u obliku proširenog informacijskog filtra, nederivacijskog informacijskog filtra te čestičnog filtra. Nadalje, također se predlaže rješenje za fuziju proizvoljnih filtara te se testira na primjeru hijerarhijske fuzije dvaju različitih tipova filtara. Glavni doprinos ovoga članka je u sustavnoj komparativnoj studiji Bayesovih metoda fuzije te u metodi za hijerarhijsku fuziju proizvoljnih filtara. Metode fuzije provjerene su na, iz višestrukih Monte Carlo simulacija dobivenom, sintetičkom skupu podataka praćenja gibajućeg objekta s više senzora različitih preciznosti analizirajući kvadratičnu Rényijevu entropiju i srednju kvadratičnu pogrešku.

Ključne riječi: Bayesova fuzija senzora, informacijski filtar, čestični filtar, Rényijeva entropija

1 INTRODUCTION

The prospects of utilizing measurements from several sensors to infer about a system state are manyfold. To begin with, the use of multiple sensors results in increased sensor measurement accuracy, and moreover, additional sensors will never reduce the performance of the optimal estimator [1]. However, in order to ensure this performance, special care must be taken when choosing the process model [2]. Furthermore, system reliability increases with additional sensors, since the system itself becomes more resilient to sensor failure [3]. Therefore, by combining data from multiple sensors, and perhaps related information from associated databases, we can achieve im-

proved accuracies and more specific inferences than using only a single sensor [4].

Sensor measurements may be combined, or fused, at a variety of levels; from the raw data level to a state vector level, or at the decision level [4]. Raw sensor data can be directly combined if the sensor data are commensurate (i.e., if the sensors are measuring the same physical phenomena), while if the sensor data are noncommensurate, then the sensor data, i.e. sensor information, must be fused at a feature/state vector level or decision level.

Information from multiple sensors can be classified as redundant or complementary [3]. *Redundant information* is provided from multiple sensors (or a single sensor over time) when each sensor perceives the same feature in the environment. On the other hand, *complementary information* from multiple sensors enables the system to perceive features impossible to perceive by using just a single sensor. But what is in common for both classifications, is that all the sensors are used to somehow infer about a system state. It is important to note that complementary sensors do not have to necessarily provide information about the full system state. Some sensors, like omnidirectional cameras and microphone arrays, measure angle and not the range of the detected objects, while laser range scanners and depth cameras can give measurements in 2D or 3D. Moreover, some sensors can provide measurements at higher rates than others, thus making sensor fusion an even more challenging problem.

One way of approaching the problem of sensor fusion is at the likelihood level. Basically, each sensor measurement is modelled as a Gaussian random variable and the resulting fused distribution is also Gaussian with the new fused mean and covariance. In [5], the fused moments are calculated by optimizing a weighted sum of Gaussian random variables so as to minimise the volume of the fused uncertainty ellipsoid. The resulting moments are equal to as if they were obtained by calculating the product of Gaussian distributions. Similar results were obtained in [6] where the fused moments are calculated by estimating the moments of a product of Gaussians via maximum likelihood approach. Both of these methods do not take any past measurements into account, and if tracking is needed then a different approach needs to be utilised.

If the system is linear and the system state is modeled as Gaussian, then multisensor fusion can be performed with the *decentralized Kalman filter* (DKF) proposed in [7]. The DKF enables us to fuse not only the measurements, but also the local independent Kalman filters. The inverse covariance form is utilized, thus resulting in additive fusion equations, which can further be elegantly translated to the *information filter* form as shown in [8]. For the case of non-linear systems the *extended information filter* (EIF) or the *unscented information filter* (UIF) [9] can be utilised. Another approach, proposed in [10], is to define for each sensor system, a separate and specific Gaussian probability distribution and to fuse them using *covariance intersection* method [11]. If the underlying distribution characterizing the system is not Gaussian and possibly non-linear, then usually *particle filters* (PF) are utilized. In [12] a distributed particle filtering algorithm is proposed where each sensor maintains a particle filter and the information is propagated in a sensor network in the form of partial likelihood functions. The last sensor then backpropagates the final importance distribution so that a new set of particles is generated at each sensor using the final distribution. The standard particle filter algorithm was decentralized in [13] by communicating and fusing only the most informative subsets of samples. It was applied on mobile robots playing the game of laser tag. In [14] a speaker tracking system was implemented by using a camera and a microphone array. Each sensor estimate was modeled as a Gaussian distribution in order to obtain the overall likelihood function. The fusion was performed by a global particle filter which used the sum of the former Gaussians as the proposal distribution and their product as the likelihood function for calculating the weights of particles.

In order to perform fusion between decentralized tracking filters, we have to take into account the common information that the distributions might share. This usually entails a product and a division of particle sets and a solution for consistent fusion was proposed in [15, 16]. An overview of decentralized fusion methods and non-Gaussian estimation techniques can be found in [17, 18]. In [19] we implicitly used centralized independent likelihood fusion via joint probabilistic data association filter in the problem of multi-target tracking with multiple sensors on a mobile robot.

In this paper we study the problem of Bayesian sensor fusion for dynamic object tracking focusing mainly on the Bayesian fusion at the likelihood and state vector level. We analyze two groups of data fusion methods: centralized independent likelihood fusion, where the sensors report only its measurement to the fusion center, and hierarchical fusion, where each sensor runs its own local estimate which is then communicated to the fusion center along with the corresponding uncertainty. Both approaches are compared and explicit solutions are presented in the form of extended information filter, unscented information filter and particle filter. Furthermore, we also propose a solution for fusion of arbitrary filters and test it on a hierarchical fusion example of two of the aforementioned filters. The main contributions of this paper are a systematic comparative study of Bayesian fusion methods, and a method for hierarchical fusion of arbitrary filters. The results are tested on a synthetic data experiment of tracking a dynamic object with several sensors of different accuracies by analyzing the quadratic

Rényi entropy and root-mean-square error.

In Section 2 we start with the mathematical background for Bayesian sensor fusion approaches where we assume that we have a centralized processor in charge of fusion from various sensor modalities. These approaches involve system dynamics modeling and state estimation, where we focus mainly on centralized independent likelihood fusion (each sensor modality reports only its likelihood), and hierarchical fusion (each sensor reports its estimated state and uncertainty). In Sections 3 and 4 we give explicit solutions to the problems in forms of extended information filter, unscented information filter and particle filter. Furthermore, we propose a solution for arbitrary filter fusion, i.e. how to fuse estimates from a Kalman and a particle filter. Moreover, we also present a solution for the case of asynchronous data arrival. In Section 5 the methods are tested and analyzed with filter entropy and tracking error on a synthetic data experiment involving dynamic object tracking, while Section 6 concludes the paper.

2 MATHEMATICAL BACKGROUND OF BAYESIAN SENSOR FUSION

The most fundamental approaches to sensor fusion were based on modeling each measurement as a Gaussian random variable, where the results were obtained with *geometrical redundant fusion* method proposed in [5] and by *maximum likelihood estimation* approach derived in [6]. The result would be a new Gaussian with an updated mean and covariance. However, it is important to note that these approaches do not take into account any past system states and depend completely on the current sensors measurements and their likelihoods.

In the present paper, the goal of sensor fusion is to estimate the system state $\boldsymbol{x}_k$ at time k based on all previous control inputs $\boldsymbol{u}_{1:k}$, and all previous sensor measurements from all the m available sensors $\boldsymbol{z}_{1:k}^{1:m}$. In other words, from a probabilistic perspective, we need to estimate the posterior distribution $p(\boldsymbol{x}_k|\boldsymbol{u}_{1:k}, \boldsymbol{z}_{1:k}^1, \boldsymbol{z}_{1:k}^2, \ldots, \boldsymbol{z}_{1:k}^m) = p(\boldsymbol{x}_k|\boldsymbol{u}_{1:k}, \boldsymbol{z}_{1:k}^{1:m})$. By applying the Bayes theorem, we can reformulate the problem as follows (for convenience we drop the condition on $\boldsymbol{u}_{1:k}$ since in tracking this is usually not known) [18]:

$$\begin{aligned} p(\boldsymbol{x}_k|\boldsymbol{z}_{1:k}^{1:m}) &= p(\boldsymbol{x}_k|\boldsymbol{z}_k^{1:m}, \boldsymbol{z}_{1:k-1}^{1:m}) \\ &= \frac{p(\boldsymbol{z}_k^{1:m}|\boldsymbol{x}_k, \boldsymbol{z}_{1:k-1}^{1:m}) p(\boldsymbol{x}_k|\boldsymbol{z}_{1:k-1}^{1:m})}{p(\boldsymbol{z}_k^{1:m}|\boldsymbol{z}_{1:k-1}^{1:m})}. \end{aligned} \quad (1)$$

Furthermore, we assume that (i) given the state $\boldsymbol{x}_k$ the measurement at the i^{th} sensor is independent of the measurements obtained from other sensors, and (ii) that the current state $\boldsymbol{x}_k$ includes all the required information to evaluate the likelihood meaning that we can drop the conditional dependency of the current measurement of the i^{th} sensor $\boldsymbol{z}_k^i$ on all the previous measurements of all the sensors $\boldsymbol{z}_{1:k-1}^{1:m}$:

$$\begin{aligned} p(\boldsymbol{z}_k^{1:m}|\boldsymbol{x}_k, \boldsymbol{z}_{1:k-1}^{1:m}) &= \prod_{i=1}^{m} p(\boldsymbol{z}_k^i \,|\, \boldsymbol{x}_k, \boldsymbol{z}_{1:k-1}^{1:m}) \\ &= \prod_{i=1}^{m} p(\boldsymbol{z}_k^i \,|\, \boldsymbol{x}_k). \end{aligned} \quad (2)$$

At this point, we can proceed further in three different directions: (i) centralized independent likelihood fusion, (ii) hierarchical fusion without feedback, and (iii) hierarchical fusion with feedback. If each sensor reports only its measurement modeled in a probabilistic manner, i.e. likelihood or the sensor model, then this leads us to *the first solution*, in which we have a global estimate of the system state updated by fusing only the likelihoods communicated from each sensor:

$$p(\boldsymbol{x}_k|\boldsymbol{z}_{1:k}^{1:m}) \propto p(\boldsymbol{x}_k|\boldsymbol{z}_{1:k-1}^{1:m}) \prod_{i=1}^{m} p(\boldsymbol{z}_k^i \,|\, \boldsymbol{x}_k), \quad (3)$$

where $p(\boldsymbol{z}_k^{1:m}|\boldsymbol{z}_{1:k-1}^{1:m})$ is omitted since it only accounts for the normalization of the calculated posterior. This is an example of *centralized independent likelihood fusion*.

Now, *the second solution* amounts to each sensor modality estimating its own local system state based only on its local observations. These local posterior estimates are then fused on a global level. Since all sensors operate without having any knowledge of other sensor measurements, at each sensor i we have $p(\boldsymbol{x}_t|\boldsymbol{z}_{1:k}^i)$ as the local posterior. By inspecting (3) we can see that we need to 'extract' the likelihood $p(\boldsymbol{z}_k^i|\boldsymbol{x}_k)$ from the local posterior. By using a similar procedure as in (1) we can derive the expression for the needed likelihood:

$$p(\boldsymbol{z}_k^i|\boldsymbol{x}_k) \propto \frac{p(\boldsymbol{x}_t|\boldsymbol{z}_{1:k}^i)}{p(\boldsymbol{x}_t|\boldsymbol{z}_{1:k-1}^i)}. \quad (4)$$

This leads us to the following expression:

$$p(\boldsymbol{x}_k|\boldsymbol{z}_{1:k}^{1:m}) \propto p(\boldsymbol{x}_k|\boldsymbol{z}_{1:k-1}^{1:m}) \prod_{i=1}^{m} \frac{p(\boldsymbol{x}_k|\boldsymbol{z}_{1:k}^i)}{p(\boldsymbol{x}_k|\boldsymbol{z}_{1:k-1}^i)}. \quad (5)$$

This is an example of *hierarchical fusion without feedback* which suggests that if we want to fuse a global prediction based on all the sensors $p(\boldsymbol{x}_k|\boldsymbol{z}_{1:k-1}^{1:m})$ with local independent sensor posteriors $p(\boldsymbol{x}_k|\boldsymbol{z}_{1:k}^i)$, we need to first remove the local prediction $p(\boldsymbol{x}_k|\boldsymbol{z}_{1:k-1}^i)$, i.e. the local prior knowledge, by a division. This is logical since we already have all the prior knowledge in the global prediction $p(\boldsymbol{x}_k|\boldsymbol{z}_{1:k-1}^{1:m})$ and are only interested in acquiring new knowledge arising from new measurements. If the local predictions $p(\boldsymbol{x}_k|\boldsymbol{z}_{1:k-1}^i)$ shared common or very similar prior information and were not removed during the fusion,

each of them would implicitly through $p(\boldsymbol{x}_k|\boldsymbol{z}^i_{1:k})$ count with each multiplication, thus resulting in a posterior being too confident, or swayed, by all the multiply-counted prior information.

For *the third solution* we have the global prediction based on all the measurements communicated back to each sensor i to serve as a new local prior which will then be updated only with the local measurement $\boldsymbol{z}^i_k$. Therefore, at each sensor i we have $p(\boldsymbol{x}_k|\boldsymbol{z}^{1:m}_{1:k-1}, \boldsymbol{z}^i_k)$ as the local posterior from which we will need to 'extract' the likelihood $p(\boldsymbol{z}^i_k|\boldsymbol{x}_k)$. Again, by following a similar procedure as in (1) we calculate the needed expression for the likelihood:

$$p(\boldsymbol{z}^i_k|\boldsymbol{x}_k) \propto \frac{p(\boldsymbol{x}_k|\boldsymbol{z}^{1:m}_{1:k-1})}{p(\boldsymbol{x}_k|\boldsymbol{z}^{1:m}_{1:k-1}, \boldsymbol{z}^i_k)}, \tag{6}$$

which leads us to the following equation for the *hierarchical fusion with feedback*:

$$p(\boldsymbol{x}_k|\boldsymbol{z}^{1:m}_{1:k}) \propto p(\boldsymbol{x}_k|\boldsymbol{z}^{1:m}_{1:k-1}) \prod_{i=1}^{m} \frac{p(\boldsymbol{x}_k|\boldsymbol{z}^{1:m}_{1:k-1}, \boldsymbol{z}^i_k)}{p(\boldsymbol{x}_k|\boldsymbol{z}^{1:m}_{1:k-1})}. \tag{7}$$

Each approach has its benefits. The centralized independent likelihood fusion is quite elegant since we only need to communicate the likelihoods to the fusion center, thus requiring only that each likelihood represents each sensor measurements faithfully. The hierarchical approach without feedback requires each sensor to run its own local estimate independent of other sensors, while the hierarchical fusion with feedback takes one step further and communicates the global fused posterior back to each sensor to serve as the next prior in the local estimation process. In this way each sensor benefits by having the same global prior, even in situations when the sensor itself has no measurements. This approach is closely related to decentralized systems where we could have several independent agents exchanging estimations in an unstructured or arbitrary network, but without central fusion processor. Although decentralization has many advantages [7, 8], it requires dealing with delayed and asequent observations, and filtering of previously exchanged common information which is a much broader topic and out of the scope of this paper. In this paper we shall concentrate on the centralized independent likelihood fusion and the hierarchical fusion without feedback, since we want to explore the effects of fusion of sensor modalities which share no common information.

2.1 Mathematical model of the tracked object

Let matrices $\mathbf{A}_k$, $\mathbf{B}_k$ and $\mathbf{G}_k$ define the propagation of the systems state, $\mathbf{H}_k$ be the observation matrix, $\boldsymbol{z}_k$ be the sensor measurement, and $\boldsymbol{x}_k$ be the states associated with the system, then the state transition and observation equations can be written as:

$$\boldsymbol{x}_k = \mathbf{A}_k\boldsymbol{x}_{k-1} + \mathbf{B}_k\boldsymbol{u}_k + \mathbf{G}_k\boldsymbol{v}_k, \tag{8}$$

$$\boldsymbol{z}_k = \mathbf{H}_k\boldsymbol{x}_k + \boldsymbol{n}_k, \tag{9}$$

where $\boldsymbol{v}_k$ and $\boldsymbol{n}_k$ describe uncertainty in the evolution of the system state and uncertainty in the measurement, respectively. Naturally, both of them are assumed to be normal, zero-mean and white. The associated local process noise and measurement noise covariance matrices are denoted as $\mathbf{Q}_k$ and $\mathbf{R}_k$.

In the present paper we use a fairly general piecewise constant white acceleration model in order to describe the system behavior [20]. The system state is defined as a vector $\boldsymbol{x}_k = [x_k, \dot{x}_k, y_k, \dot{y}_k]$, where (x_k, y_k) are the Cartesian coordinates, while $\dot{x}_k$ and $\dot{y}_k$ represent their respective velocities in the x, y-plane. The model itself is given by:

$$\begin{aligned}\boldsymbol{x}_k = \mathbf{A}\boldsymbol{x}_{k-1} + \mathbf{G}\boldsymbol{v}_k = &\begin{bmatrix} 1 & \Delta T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \Delta T \\ 0 & 0 & 0 & 1 \end{bmatrix} \boldsymbol{x}_{k-1} \\ &+ \begin{bmatrix} \frac{\Delta T^2}{2} & 0 \\ \Delta T & 0 \\ 0 & \frac{\Delta T^2}{2} \\ 0 & \Delta T \end{bmatrix} \boldsymbol{v}_k,\end{aligned} \tag{10}$$

where ΔT is the sampling period.

For the measurement model, we assume that the sensors measure both range and bearing, thus yielding a nonlinear measurement equation:

$$\boldsymbol{z}_k = \boldsymbol{h}\left(\boldsymbol{x}_k\right) + \boldsymbol{n}_k = \begin{bmatrix} \sqrt{x_k^2 + y_k^2} \\ \arctan\left(\dfrac{y_k}{x_k}\right) \end{bmatrix} + \boldsymbol{n}_k. \tag{11}$$

3 CENTRALIZED SENSOR FUSION

3.1 Centralized extended information filter

With transition and observation equations defined with (10) and (11), respectively, for the Kalman filter the *a priori* predicted values of the system state and covariance are calculated as follows:

$$\hat{\boldsymbol{x}}_{k|k-1} = \mathbf{A}_k\hat{\boldsymbol{x}}_{k-1|k-1} + \mathbf{B}_k\boldsymbol{u}_k, \tag{12}$$

$$\mathbf{P}_{k|k-1} = \mathbf{A}_k\mathbf{P}_{k-1|k-1}\mathbf{A}_k^T + \mathbf{G}_k\mathbf{Q}_k\mathbf{G}_k^T. \tag{13}$$

Instead of continuing with the Kalman filter update equations, we shall now revert to its equivalent information filter form, whose advantages in sensor fusion will become apparent soon.

The information matrix $\mathbf{Y}_{k|k}$ and the information vector $\hat{\boldsymbol{y}}_{k|k}$ are defined as follows:

$$\mathbf{Y}_{k|k} = \mathbf{P}_{k|k}^{-1}, \quad \hat{\boldsymbol{y}}_{k|k} = \mathbf{P}_{k|k}^{-1}\hat{\boldsymbol{x}}_{k|k}. \tag{14}$$

If we define the information associated with the observation taken at time k as:

$$\boldsymbol{i}_k = \mathbf{H}_k^{\mathrm{T}}\mathbf{R}_k^{-1}\left(\boldsymbol{\nu}_k + \mathbf{H}_k\hat{\boldsymbol{x}}_{k|k-1}\right), \quad \mathbf{I}_k = \mathbf{H}_k^{\mathrm{T}}\mathbf{R}_k^{-1}\mathbf{H}_k, \tag{15}$$

where $\boldsymbol{\nu}_k = \boldsymbol{z}_k - \boldsymbol{h}(\hat{\boldsymbol{x}}_{k|k-1})$ is the innovation vector, we can write the update stage of the information filter as:

$$\hat{\boldsymbol{y}}_{k|k} = \hat{\boldsymbol{y}}_{k|k-1} + \boldsymbol{i}_k, \tag{16}$$

$$\mathbf{Y}_{k|k} = \mathbf{Y}_{k|k-1} + \mathbf{I}_k. \tag{17}$$

From (16) and (17) we can see that the update stage of the information filter is additive. In fact, this very property of the information filter is the main reason for its utility in multisensor fusion.

If we have N sensors, then for each sensor i we can define an observation equation:

$$\boldsymbol{z}_k^i = \mathbf{H}_k^i\boldsymbol{x}_k + \boldsymbol{n}_k^i, \quad i = 1, \ldots, N. \tag{18}$$

Since the measurement model can be linearised about the predicted state vector, the Jacobian matrix may be introduced:

$$\mathbf{H}_k^i = \left.\frac{\partial \boldsymbol{h}^i(\boldsymbol{x})}{\partial \boldsymbol{x}}\right|_{\mathbf{x}=\hat{\mathbf{x}}_{k|k-1}}. \tag{19}$$

For the measurement model (11) the Jacobian matrix takes the following form:

$$\frac{\partial \boldsymbol{h}(\boldsymbol{x})}{\partial \boldsymbol{x}} = \begin{bmatrix} \dfrac{x}{\sqrt{x^2+y^2}} & 0 & \dfrac{y}{\sqrt{x^2+y^2}} & 0 \\ -\dfrac{y}{x^2+y^2} & 0 & \dfrac{x}{x^2+y^2} & 0 \end{bmatrix}. \tag{20}$$

In the standard Kalman filter notation, contributions from multiple sensors cannot be additively combined, since, although the sensor measurements given the system state are themselves independent, the innovations are correlated through common information from the prediction stage in (12). However, in the information form, the terms $\boldsymbol{i}_k^i$ from each sensor are uncorrelated, thus resulting with additive update stage with contributions from each sensor [7, 8]:

$$\hat{\boldsymbol{y}}_{k|k} = \hat{\boldsymbol{y}}_{k|k-1} + \sum_{i=1}^{N} \boldsymbol{i}_k^i, \tag{21}$$

$$\mathbf{Y}_{k|k} = \mathbf{Y}_{k|k-1} + \sum_{i=1}^{N} \mathbf{I}_k^i, \tag{22}$$

where now $\hat{\boldsymbol{y}}_{k|k}$ and $\mathbf{Y}_{k|k}$ represent the central fused information vector and information matrix. The central fused estimate of the system state may be found via $\hat{\boldsymbol{x}}_{k|k} = \mathbf{Y}_{k|k}^{-1}\hat{\boldsymbol{y}}_{k|k}$.

By inspecting (21) and (15), we can see that during fusion each sensor measurement is weighted by its corresponding variance. In essence, this approach is similar to the product of Gaussians, except that it does take past values into account through $\hat{\mathbf{y}}_{k|k-1}$, which we can see from (14) that it is just the predicted global system state weighted by the corresponding global predicted variance.

The previous approach to sensor fusion was derived in [7] following the work in [21], and was termed *decentralized Kalman filter* (DKF). The main idea was to offer a flexible method for decomposing the linear Kalman filter into autonomous local processors associated with each sensor modality. However, so far we have presented only the means for fusing multiple sensor measurements. If we want to fuse estimates form several running filters each adjoined to a sensor (for which the DKF was initially derived for), then we have to further extend the fusion approach.

3.2 Centralized unscented information filter

In this section the unscented version [22, 23] of the information filter is utilized for centralized sensor fusion. Unlike EIF which approximates the non-linear function by a Taylor series expansions, the UIF deterministically generates *sigma points* and uses them to estimate the mean and the covariance. Therefore, for an n dimensional system we need to generate $2n+1$ sigma points $\boldsymbol{\mathcal{X}}_{j,k-1}$ by:

$$\begin{aligned} \boldsymbol{\mathcal{X}}_{0,k-1|k-1} &= \hat{\boldsymbol{x}}_{k-1|k-1} \\ \boldsymbol{\mathcal{X}}_{j,k-1|k-1} &= \hat{\boldsymbol{x}}_{k-1|k-1} + \left(\sqrt{(n+\lambda)\,\mathbf{P}_{k-1|k-1}}\right)_j \\ \boldsymbol{\mathcal{X}}_{j,k-1|k-1} &= \hat{\boldsymbol{x}}_{k-1|k-1} - \left(\sqrt{(n+\lambda)\,\mathbf{P}_{k-1|k-1}}\right)_j, \end{aligned} \tag{23}$$

where $\lambda = \alpha^2(n+\kappa) - n$ is a scaling parameter with $0 \leq \alpha \leq 1$ and κ usually chosen by the heuristic $n + \kappa = 3$, and $\left(\sqrt{(n+\lambda)\,\mathbf{P}_{k-1|k-1}}\right)_j$ is the j^{th} column of the square root matrix of the multiplied covariance matrix.

The corresponding weights for recovering the mean and the covariance are calculated as follows:

$$\begin{aligned} w_0^{(m)} &= \lambda/(n+\lambda) \\ w_j^{(m)} &= 1/\left[2(n+\lambda)\right] \\ w_0^{(c)} &= \lambda/(n+\lambda) + (1-\alpha^2+\beta) \\ w_j^{(c)} &= 1/\left[2(n+\lambda)\right], \end{aligned} \tag{24}$$

where the parameter β is for encoding additional higher order effects. If the underlying distribution is a Gaussian, then $\beta = 2$ is the optimal choice.

The information prediction equations are:

$$\hat{\boldsymbol{y}}_{k|k-1} = \mathbf{Y}_{k|k-1} \sum_{j=0}^{2n} w_j^{(m)} \boldsymbol{\mathcal{X}}_{j,k|k-1}, \tag{25}$$

$$\mathbf{Y}_{k|k-1} = \mathbf{P}_{k|k-1}^{-1}, \tag{26}$$

where $\boldsymbol{\mathcal{X}}_{j,k|k-1}$ are predicted sigma points calculated by the process model (10), and the predicted covariance matrix is computed by:

$$\begin{aligned}\mathbf{P}_{k|k-1} = &\sum_{j=0}^{2n} w_j^{(c)} \left[\boldsymbol{\mathcal{X}}_{j,k|k-1} - \hat{\boldsymbol{x}}_{k|k-1}\right] \\ &\cdot \left[\boldsymbol{\mathcal{X}}_{j,k|k-1} - \hat{\boldsymbol{x}}_{k|k-1}\right]^T + \mathbf{G}\mathbf{Q}_k\mathbf{G}^T.\end{aligned} \tag{27}$$

In order to present the UIF update equations, let us first define a pseudo measurement matrix $\boldsymbol{\mathcal{H}}_k$ as [9]:

$$\boldsymbol{\mathcal{H}}_k^T = \mathbf{P}_{k|k-1}^{-1} \mathbf{P}_{k|k-1}^{\mathcal{X},\mathcal{Z}}, \tag{28}$$

where the cross-covariance matrix is calculated by:

$$\begin{aligned}\mathbf{P}_{k|k-1}^{\mathcal{X},\mathcal{Z}} = &\sum_{j=1}^{2n} w_j^{(c)} \left[\boldsymbol{\mathcal{X}}_{j,k|k-1} - \hat{\boldsymbol{x}}_{k|k-1}\right] \\ &\cdot \left[\boldsymbol{\mathcal{Z}}_{j,k|k-1} - \hat{\boldsymbol{z}}_{k|k-1}\right]^T,\end{aligned} \tag{29}$$

where $\boldsymbol{\mathcal{Z}}_{j,k|k-1} = \boldsymbol{h}\left(\boldsymbol{\mathcal{X}}_{j,k|k-1}\right)$ are observation sigma points, and the predicted measurement vector is obtained by $\hat{\boldsymbol{z}}_{k|k-1} = \sum_{j=0}^{2n} w_j^{(m)} \boldsymbol{\mathcal{Z}}_{j,k|k-1}$. Then, in terms of psedo-measurement matrix, information contribution for sensor i can be expressed as [1]:

$$\boldsymbol{i}_k^i = \boldsymbol{\mathcal{H}}_{i,k}^T \mathbf{R}_{i,k}^{-1} \left[\boldsymbol{z}_k^i - \hat{\boldsymbol{z}}_{k|k-1} + \boldsymbol{\mathcal{H}}_{i,k}\hat{\boldsymbol{x}}_{k|k-1}\right], \tag{30}$$

$$\mathbf{I}_k^i = \boldsymbol{\mathcal{H}}_{i,k}^T \mathbf{R}_{i,k}^{-1} \boldsymbol{\mathcal{H}}_{i,k}. \tag{31}$$

Now, the measurements are fused just as in the case of EIF, through (21) and (22).

3.3 Centralized particle filter

In the previous sections we have focused mainly on filters which assume unimodal (Gaussian) distribution over the system state. In many applications this assumption may not be adequate and more versatile representations may be needed. In this section we present methods for sensor fusion via particle filters which due to their specific representation of density need additional tools to calculate the update equations.

Let $\{\boldsymbol{x}^p, w^p\}_{p=1}^P$ denote a random measure that characterises the posterior pdf $p(\boldsymbol{x})$, where $\{\boldsymbol{x}^p,\ p = 1, \ldots, P\}$ is a set of particles with associated weights $\{w^p,\ p = 1, \ldots, P\}$. The weights are normalised so that $\sum_p w^p = 1$. Then, the posterior density can be approximated as [24, 25]:

$$p(\boldsymbol{x}_k) \approx \sum_{p=1}^{P} w_k^p \delta(\boldsymbol{x}_k - \boldsymbol{x}_k^p), \tag{32}$$

where P is the number of particles and $\delta(.)$ is the Dirac delta measure. The expectation of some function $f(\boldsymbol{x})$ integrable with respect to the pdf $p(\boldsymbol{x})$ is:

$$E\left[f(\boldsymbol{x})\right] = \int f(\boldsymbol{x}) p(\boldsymbol{x})\, d\boldsymbol{x}, \tag{33}$$

and the approximation of the integral with particles is:

$$E\left[f(\boldsymbol{x})\right] \approx \frac{1}{P} \sum_{p=1}^{P} f(\boldsymbol{x}^p), \tag{34}$$

where $\boldsymbol{x}^p \sim p(\boldsymbol{x})$ and the expectation converges to the true values as $P \to \infty$. Often, it is hard to sample from the true distribution, hence *importance sampling* is used. The main idea is to sample from a proposal distribution $q(\boldsymbol{x})$ which encompassed the support space of $p(\boldsymbol{x})$, and then we can rewrite (33) as:

$$E\left[f(\boldsymbol{x})\right] = \int f(\boldsymbol{x}) \frac{p(\boldsymbol{x})}{q(\boldsymbol{x})} q(\boldsymbol{x})\, d\boldsymbol{x} = \int f(\boldsymbol{x}) w(\boldsymbol{x}) q(\boldsymbol{x})\, d\boldsymbol{x}, \tag{35}$$

where the *importance weights* $w(\boldsymbol{x})$ is given as $w(\boldsymbol{x}) = p(\boldsymbol{x})/q(\boldsymbol{x})$. An estimate of the expectation is then given by:

$$E\left[f(\boldsymbol{x})\right] \approx \frac{1}{P} \sum_{p=1}^{P} f(\boldsymbol{x}^p) w(\boldsymbol{x}^p). \tag{36}$$

In the centralized solution all the sensor modalities report only their measurements (likelihoods), which corresponds to estimating the posterior via (3). After similar derivation to the one in [24] we obtain the expression for weights calculation:

$$w(\boldsymbol{x}_k^p) \propto w(\boldsymbol{x}_{k-1}^p) \frac{p(\boldsymbol{x}_k^p | \boldsymbol{x}_{k-1}^p)}{q(\boldsymbol{x}_k^p | \boldsymbol{x}_{k-1}^p, \boldsymbol{z}_k^{1:m})} \prod_{i=1}^{m} p(\boldsymbol{z}_k^i \,|\, \boldsymbol{x}_k) \tag{37}$$

If we choose the prior as the proposal density, $q(\boldsymbol{x}_k^p | \boldsymbol{x}_{k-1}^p, \boldsymbol{z}_k^{1:m}) = p(\boldsymbol{x}_k^p | \boldsymbol{x}_{k-1}^p)$, then weights are calculated from the following expression:

$$w(\boldsymbol{x}_k) \propto w(\boldsymbol{x}_{k-1}^p) \prod_{i=1}^{m} p(\boldsymbol{z}_k^i \,|\, \boldsymbol{x}_k), \tag{38}$$

[1] Here we use index i in matrices $\boldsymbol{\mathcal{H}}_{i,k}$ and $\mathbf{R}_{i,k}$ to denote the sensor i in the subscript instead of superscript in order to more clearly denote the transpose and the inverse operators.

where the sensor likelihood is defined by:

$$p(\boldsymbol{z}_k^i \,|\, \boldsymbol{x}_k) = \frac{1}{2\pi\sqrt{|\mathbf{R}_{i,k}|}} \cdot \exp\left\{-\frac{1}{2}\left[\boldsymbol{z}_k^i - \boldsymbol{h}^i(\boldsymbol{x}_k)\right]^{\mathrm{T}} \mathbf{R}_{i,k}^{-1}\left[\boldsymbol{z}_k^i - \boldsymbol{h}^i(\boldsymbol{x}_k)\right]\right\}. \tag{39}$$

Once the weights are calculated we can estimate the state as follows:

$$\hat{\boldsymbol{x}}_{k|k} = E[\boldsymbol{x}_k|\boldsymbol{z}_k] \approx \frac{1}{P}\sum_{p=1}^{P} w(\boldsymbol{x}_k^p)\boldsymbol{x}_k^p. \tag{40}$$

The resampling of the particles is done at each iteration via the *sequential importance resampling* (SIR) algorithm [24]. At this point, also sample size adaption could be performed [26]. Concerning the prediction stage of the filter, we use the model (10) to predict the state of each particle.

4 HIERARCHICAL SENSOR FUSION

4.1 Hierarchical fusion of information filters

In this example each sensor runs its own local instance of the EIF; prediction through (12) and (13), and update through (16) and (17). Furthermore, all sensor modalities utilise the same process model (10). The central processor, on the other hand, also runs its own instance of EIF; prediction through (12) and (13) with the same process model (10) as the sensors utilise, but the global update, i.e. fusion, should be performed in the following manner [8]:

$$\hat{\boldsymbol{y}}_{k|k} = \hat{\boldsymbol{y}}_{k|k-1} + \sum_{i=1}^{N}\left[\hat{\boldsymbol{y}}_{i,k|k} - \hat{\boldsymbol{y}}_{i,k|k-1}\right], \tag{41}$$

$$\mathbf{Y}_{k|k} = \mathbf{Y}_{k|k-1} + \sum_{i=1}^{N}\left[\mathbf{Y}_{i,k|k} - \mathbf{Y}_{i,k|k-1}\right]. \tag{42}$$

We can see that the sensor modalities only have to communicate the difference between the updated, $\hat{\boldsymbol{y}}_{i,k|k}$, and the predicted, $\hat{\boldsymbol{y}}_{i,k|k-1}$, information vector. The same applies for the update of the information matrix. This ensures that only the new information is used for fusion.

Hierarchical sensor fusion with UIF is performed in a similar manner as with EIF. Both sensor modalities run their own local, independent, and autonomous UIF and report their estimates to the central fusion processor. The central processor runs a global UIF, and performs the global update, i.e. fusion, through (41) and (42).

4.2 Hierarchical sensor fusion with particle filtering

In this hierarchical solution with particle filters each sensor modality runs its own local independent particle filter, which needs to be fused with the global particle filter. This corresponds to estimating the posterior via (5). Therefore, the importance weights are given by:

$$w(\boldsymbol{x}_k^p) \propto w(\boldsymbol{x}_{k-1}^p)\frac{p(\boldsymbol{x}_k^p|\boldsymbol{x}_{k-1}^p)}{q(\boldsymbol{x}_k^p|\boldsymbol{x}_{k-1}^p, \boldsymbol{z}_k^{1:m})}\prod_{i=1}^{m}\frac{p(\boldsymbol{x}_k|\boldsymbol{z}_{1:k}^i)}{p(\boldsymbol{x}_k|\boldsymbol{z}_{1:k-1}^i)}. \tag{43}$$

If we again choose the global prior as the proposal density, $q(\boldsymbol{x}_k^p|\boldsymbol{x}_{k-1}^p, \boldsymbol{z}_k^{1:m}) = p(\boldsymbol{x}_k^p|\boldsymbol{x}_{k-1}^p)$, then weights are calculated from the following expression:

$$w(\boldsymbol{x}_k^p) \propto w(\boldsymbol{x}_{k-1}^p)\prod_{i=1}^{m}\frac{p(\boldsymbol{x}_k|\boldsymbol{z}_{1:k}^i)}{p(\boldsymbol{x}_k|\boldsymbol{z}_{1:k-1}^i)}. \tag{44}$$

By inspecting (44) we can see that we need to explicitly calculate functions $p(\boldsymbol{x}_k|\boldsymbol{z}_{1:k}^i)$ and $p(\boldsymbol{x}_k|\boldsymbol{z}_{1:k-1}^i)$, but we only have particles estimating the mentioned densities. If all the P weights from all the m distributions were on the same support space, then explicit multiplication and division of weights would be possible. But since most weights are assigned to an infinitesimally small point mass, direct multiplication and division is not applicable. To solve this problem we need a way to estimate the density function from a particle set. One such method is the Parzen window method [27] which involves placing a kernel function on top of each sample and then evaluating the density as a sum of the kernels. We continue this approach as proposed in [17, 28], and convert each sample to a kernel:

$$K_h(\boldsymbol{x}_k) = h^n K(\boldsymbol{x}_k), \tag{45}$$

where $K(.)$ is the particle set covariance, and $h > 0$ is the scaling parameter. For the kernel, we choose:

$$h = \left(\frac{4}{n+2}\right)^{e} P^{-e}, \tag{46}$$

where $e = \frac{1}{n+4}$, and P is the number of particles. At this point, the estimated density function is described as a sum of Gaussian kernels:

$$p\left(\boldsymbol{x}_k|\,\boldsymbol{z}_{1:k}^i\right) = \sum_{p=1}^{P}\mathcal{N}(\boldsymbol{x}_k; \boldsymbol{x}_k^p, 2K_h(\boldsymbol{x}_k|\boldsymbol{z}_{1:k}^i)). \tag{47}$$

In [15, 16] the authors propose how to utilize this function estimation for particle set multiplication and division.

4.3 Hierarchical fusion of arbitrary filters

In the previous sections we have addressed centralized and hierarchical fusion of EIFs, UIFs, and PFs. But what if wee need to fuse a combination of these filters? For an example, an EIF and a PF? In this section we propose a solution to such a problem.

To answer the question, we first need to choose the global filter which will actually keep the global track and fuse the local filters. In most cases we will utilize the filter which has better or higher modeling capabilities. For an example, UIF is known to handle better high non-linearities than the EIF. Thus if fusing an UIF and an EIF, we might choose a UIF for the global filter. Furthermore, if one of the filters is a PF, then we might choose also a PF for the global filter, since it is capable of handling both non-linearities and multimodal distributions. This reasoning stems from the fact that if we are using a more versatile filter for local estimation, there must have been a good reason for such a choice, and the global filter should be equally versatile. However, this might not always be the case and there may be situations in which a less versatile and computationally less complex filter could be applied for fusion. Therefore, in this section we shall analyze both of the aforementioned situations, i.e. fusion of a local EIF and PF with a hierarchical EIF, and the fusion of local EIF and PF with a hierarchical PF.

For the case of fusion with the hierarchical EIF we have equations defined in Section 4.1, from which we can see that we need to calculate the difference of the information vectors and matrices of the local updated and predicted states. For the case of the local EIF this is straightforward, while for the case of the local PF we propose to calculate the covariance of the particle set first:

$$\begin{aligned}\mathbf{P}_{k|k} &= E[(\boldsymbol{x}_k - E[\boldsymbol{x}_k])(\boldsymbol{x}_k - E[\boldsymbol{x}_k])^{\mathrm{T}}|\boldsymbol{z}_k] \\ &\approx \frac{1}{P}\sum_{p=1}^{P} w(\boldsymbol{x}_k^p)(\boldsymbol{x}_{k|k}^p - \hat{\boldsymbol{x}}_{k|k})(\boldsymbol{x}_{k|k}^p - \hat{\boldsymbol{x}}_{k|k})^{\mathrm{T}},\end{aligned} \tag{48}$$

which can then be used to calculate the information variables via (14). Once having analogously calculated the information variables for the prediction, we can readily fuse the local PF and EIF with the hierarchical EIF via (41) and (42).

For the case of fusion with the hierarchical PF, we have presented fusion equations in Section 4.2, from which we can see that in order to calculate the weights of the hierarchical PF we need to explicitly evaluate the prior and the posterior density of the local EIF and PF. Since EIF assumes a Gaussian distribution, the updated density will have the following form:

$$\begin{aligned}p(\boldsymbol{x}_k|\boldsymbol{z}_{1:k}^i) = \mathcal{N}(\boldsymbol{x}_k; \hat{\boldsymbol{x}}_{k|k}, \mathbf{P}_{k|k}) = \frac{1}{2\pi\sqrt{|\mathbf{P}_{k|k}|}} \\ \cdot \exp\left\{-\frac{1}{2}\left[\boldsymbol{x}_k - \hat{\boldsymbol{x}}_{k|k}\right]^{\mathrm{T}} \mathbf{P}_{k|k}^{-1}\left[\boldsymbol{x}_k - \hat{\boldsymbol{x}}_{k|k}\right]\right\}.\end{aligned} \tag{49}$$

A similar expression can be obtained for the prediction $\boldsymbol{x}_{k|k-1}$ and $\mathbf{P}_{k|k-1}$. Furthermore, in order to be able to divide the prior and the posterior density of the particle filter we will need to resort to the kernel density estimation method presented in Section 4.2. Hence, the updated and the predicted densities will have the form defined in (47).

All this results with the following expression for the calculation of the weights $\boldsymbol{w}(\boldsymbol{x}_k^q)$ of the global PF which relies on the expressions derived for hierarchical sensor fusion (5) and on the calculation of the hierarchical particle filter weights (44):[2]

$$\begin{aligned}w(\boldsymbol{x}_k^q) \propto w(\boldsymbol{x}_{k-1}^q) \cdot \frac{\sum_{p=1}^{P}\mathcal{N}(\boldsymbol{x}_k^q; \boldsymbol{x}_k^p, 2K_h(\boldsymbol{x}_k|\boldsymbol{z}_{1:k}^i))}{\sum_{p=1}^{P}\mathcal{N}(\boldsymbol{x}_k^q; \boldsymbol{x}_k^p, 2K_h(\boldsymbol{x}_k|\boldsymbol{z}_{1:k-1}^i))} \\ \cdot \frac{\mathcal{N}(\boldsymbol{x}_k^q; \hat{\boldsymbol{x}}_{k|k}, \mathbf{P}_{k|k})}{\mathcal{N}(\boldsymbol{x}_k^q; \hat{\boldsymbol{x}}_{k|k-1}, \mathbf{P}_{k|k-1})}.\end{aligned} \tag{50}$$

4.4 Asynchronous fusion

In the analysis thus far, we have assumed that all the measurements/estimates arrive synchronously to the fusion center. In most real world applications this might not be the case. So, the question is, how should the fusion be calculated if the measurements/estimates arrive asynchronously? Is there a difference for the centralized and hierarchical case?

Let us assume at this point that all the sensors send their measurements/estimates in fixed, but different time intervals. For an example, if we have three sensors, two might report each 25 ms, while the third might report each 60 ms. In such a case, only the fusion center has to change in order to accommodate asynchronous arrivals, since from a local sensor's point of view, nothing has actually changed.

By inspecting (3) and (5), we see that for the sheer aspect of fusion itself, we only need to change m, the number of sensors that we are fusing at a certain point. But there is also one more very subtle change that needs to be addressed. When we use (10) for state prediction we assume that the object undergoes a constant acceleration during a given sampling period, which makes it inappropriate for asynchronous fusion where the prediction and update occurs in practically arbitrary time intervals [20]. This, in effect, changes the way we must calculate the prediction of the state, and the solution is to switch to discretized continuous white noise acceleration model.

Basically, the state prediction equation remains the same, only the process noise covariance matrix needs to

[2] Note that the particular particle in the hierarchical PF is now denoted with q instead of p in order to leave the p to denote the particles in the local PF

be evaluated differently [20]:

$$\tilde{\mathbf{Q}}_k = E\left[\boldsymbol{v}_k \boldsymbol{v}_k^{\mathrm{T}}\right] = \tilde{q}\int_0^{\Delta T} \begin{bmatrix} \Delta T-\tau & 0 \\ 1 & 0 \\ 0 & \Delta T-\tau \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \Delta T-\tau & 1 & 0 & 0 \\ 0 & 0 & \Delta T-\tau & 1 \end{bmatrix} d\tau = \tilde{q}\begin{bmatrix} \frac{1}{3}\Delta T^3 & \frac{1}{2}\Delta T^2 & 0 & 0 \\ \frac{1}{2}\Delta T^2 & \Delta T & 0 & 0 \\ 0 & 0 & \frac{1}{3}\Delta T^3 & \frac{1}{2}\Delta T^2 \\ 0 & 0 & \frac{1}{2}\Delta T^2 & \Delta T \end{bmatrix}, \tag{51}$$

where $\tilde{q}$ is the continuous-time process noise intensity assumed to be a constant. Recommendations on how to choose $\tilde{q}$ can be found in [20]. This implicitly means that $\mathbf{G}_k\mathbf{Q}_k\mathbf{G}_k^{\mathrm{T}} \rightarrow \tilde{\mathbf{Q}}_k$ in (13).

The result above suggests that regardless of the type of fusion, centralized or hierarchical, we only need to correctly calculate the process noise covariance if the measurements/estimates arrive in asynchronous, but locally fixed, time intervals.

5 EVALUATION

In this section we test the sensor fusion methods on the problem of object tracking with multiple sensors. For the purpose of simulating a moving object we used a nearly coordinated turn rate model with large process noise [20] (see APPENDIX A). This model differs intentionally from the model used in prediction which is defined in Section 2.1 since it is possible that true dynamics of the object are unknown (consider the problem of people tracking). The tracked object is observed at all time by two sensors, one being more precise than the other. The measurements of the first and the second sensor are both corrupted with white Gaussian noise. Figure 1 shows an example of the simulated trajectory and the measurements of the sensors. For the purpose of more extensive analysis we have performed 50 Monte Carlo runs, each of which consisted of randomly generated trajectories like the one presented in Fig. 1. Hereafter, we assume that only one object is being tracked and that all the sensor measurements arrive synchronously. For monitoring tracker performance we utilize entropy and root-mean-square-error (RMSE).

5.1 Entropy and RMSE

We utilize entropy $H(\boldsymbol{x}_t)$ as a measure of the tracker performance. Entropy is a very useful measure of informativeness, and therefore we use it to track 'confidence' of the tracker in its estimates. This way we can analyze how sensor fusion affects the tracker's informativeness. Ideally, by including more sensors, even the less precise ones, we should experience an increase in informativeness, i.e. a decrease in entropy.

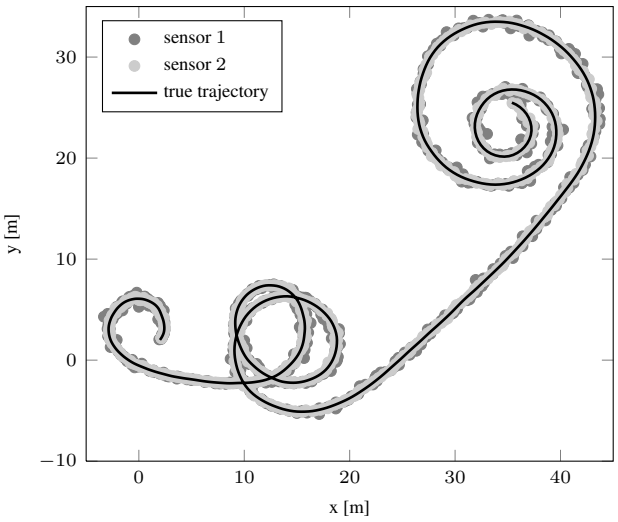

Fig. 1. An example of a simulated trajectory of a moving object and measurements of two sensors with different noise parameters

A measure of entropy can take many analytical forms. Shannon entropy can be difficult to analytically work with, e.g. Shannon entropy of a mixture of distributions cannot be expressed in closed-form, and therefore we chose to work with Rényi entropy which usually offers a more suitable framework for analytical calculations [29]. The Rényi quadratic entropy of a random variable $\boldsymbol{x}_t$ with a Gaussian distribution is given by

$$H_2(\boldsymbol{x}_t) = \frac{n}{2}\log 4\pi + \frac{1}{2}\log|\mathbf{P}_t|, \tag{52}$$

where n is the state dimension and the entropy is proportional to the logarithm of the determinant of the covariance $\mathbf{P}_t$.

Entropy calculation of continuous random variables is based on the probability density functions of these variables. In order to calculate entropy of a particle filter, which rather represents the density and not the function, we need a non-parametric method to estimate the pdf. As in Section 4.2 we will utilize the Parzen window method [27] which estimates the density as a sum of Gaussian kernels for which an analytical solution for the quadratic Rényi entropy exists [30]:

$$H_2(\boldsymbol{x}_t) = -\log\frac{1}{P^2}\sum_{i=1}^{P}\sum_{j=1}^{P}\mathcal{N}(\boldsymbol{x}_t^i - \boldsymbol{x}_t^j; 0, 2K_h(\boldsymbol{x}_t)). \tag{53}$$

Table 1. Evaluation results of the sensor fusion for 50 Monte Carlo runs (number of particles was 250)

	RMSE position [m] (velocity [m/s])		
	EIF	UIF	PF
Centralized	0.11 (0.17)	0.08 (0.17)	0.08 (0.34)
Sensor 1	0.19 (0.24)	0.13 (0.32)	0.18 (0.45)
Sensor 2	0.12 (0.18)	0.13 (0.34)	0.08 (0.26)
Hierarchical	0.11 (0.17)	0.13 (0.34)	0.09 (0.31)
Arbitrary	local EIF	local PF	fused EIF
	0.06 (0.20)	0.25 (0.47)	0.07 (0.34)
	local EIF	local PF	fused PF
	0.06 (0.20)	0.25 (0.47)	0.29 (0.60)

Table 2. Evaluation results of the sensor fusion for the given synthetic data example (number of particles was 1000)

	RMSE position [m] (velocity [m/s])		
	EIF	UIF	PF
Centralized	0.69 (1.59)	0.46 (0.85)	0.10 (0.59)
Sensor 1	1.10 (2.13)	0.23 (0.70)	0.25 (0.98)
Sensor 2	0.77 (1.72)	0.22 (0.68)	0.10 (0.66)
Hierarchical	0.70 (1.61)	0.22 (0.69)	0.13 (0.54)
Arbitrary	local EIF	local PF	fused EIF
	0.10 (0.66)	0.11 (0.66)	0.09 (0.59)
	local EIF	local PF	fused PF
	0.10 (0.66)	0.11 (0.66)	0.11 (0.60)

The RMSE is calculated both for the position and velocity as follows

$$e_{\text{pos}} = \sqrt{\frac{1}{T}\sum_{k=1}^{T}(\hat{x}_t - x_t)^2 + (\hat{y}_t - y_t)^2}$$
$$e_{\text{vel}} = \sqrt{\frac{1}{T}\sum_{k=1}^{T}(\hat{\dot{x}}_t - \dot{x}_t)^2 + (\hat{\dot{y}}_t - \dot{y}_t)^2}, \tag{54}$$

where T is the simulation length, $(\hat{x}_t, \hat{y}_t)$ are estimated coordinates and (x_t, y_t) are true coordinates at time index k, while $(\hat{\dot{x}}_t, \hat{\dot{y}}_t)$ are the estimated velocities and $(\dot{x}_t, \dot{y}_t)$ are true velocities at time index k.

5.2 Comparative analysis

In this section we will present the entropy and the RMSE for the cases of centralized, hierarchical fusion and for the examples of fusing an EIF and a PF through a global EIF and PF. Table 1 shows the the obtained position and velocity RMSE results of 50 Monte Carlo runs, where the the top and bottom performing 10% were removed and the mean was taken from the rest of the results. Table 2 shows the obtained position and velocity RMSE for the given example of the simulated trajectory. In centralized fusion all the measurements from the sensors were communicated to the fusion node which in turn ran an estimator and fused the measurements via (21) and (22), (30) and (31), and (38) for the cases of EIF, UIF, and PF, respectively. In the case of hierarchical fusion each sensor ran its own local estimator, which communicated its estimate to the fusion node, which then via (41) and (42), and (44) fused the local estimates. Figs. 2, 3 and 4 show the given example entropy of sensor 1, sensor 2, and the fused EIF,UIF and PF, respectively. For all the examples of hierarchical fusion we can notice a pattern in which the fused estimator had similar RMSE as the more precise sensor, but smaller entropy than any of local sensor estimators indicating a reduction in uncertainty as can be seen in Figs. 2, 3, and 4. This result showed that although we fused a very precise sensor with a less precise one, the resulting estimator did in fact have a benefit in form of a reduced uncertainty.

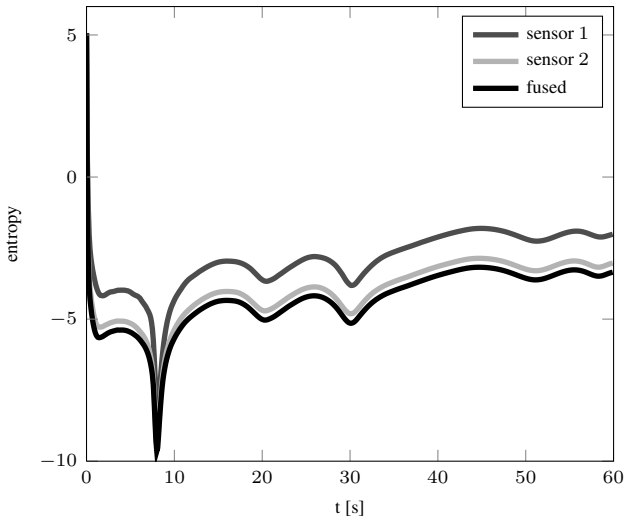

Fig. 2. Entropy of the EIF *tracker with the first sensor, with the second sensor, and the entropy of the fused hierarchical* EIF *tracker for the given synthetic example*

Another parameter that has been analyzed is the execution time [3] of each of the fusion algorithms by averaging the time of each of the 50 Monte Carlo runs. Consider-

[3]The experiments were conducted on a laptop computer with an Intel Core i7 CPU@1.60GhZ, Matlab 2010a and Ubuntu 12.04 operating system

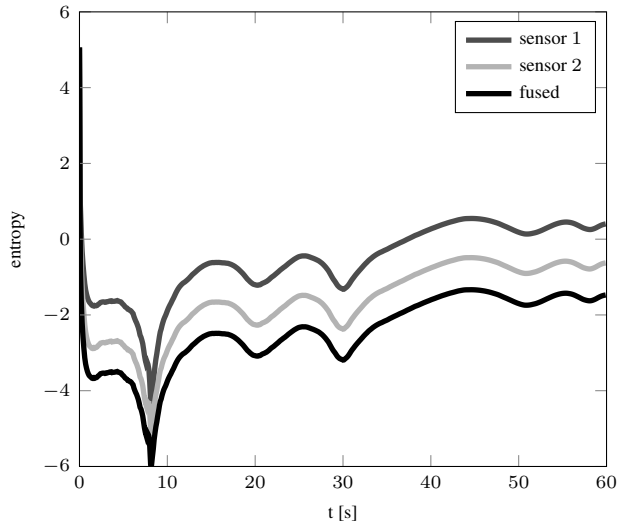

Fig. 3. Entropy of the UIF *with the first sensor, with the second sensor, and the entropy of the fused hierarchical* UIF *for the given synthetic example*

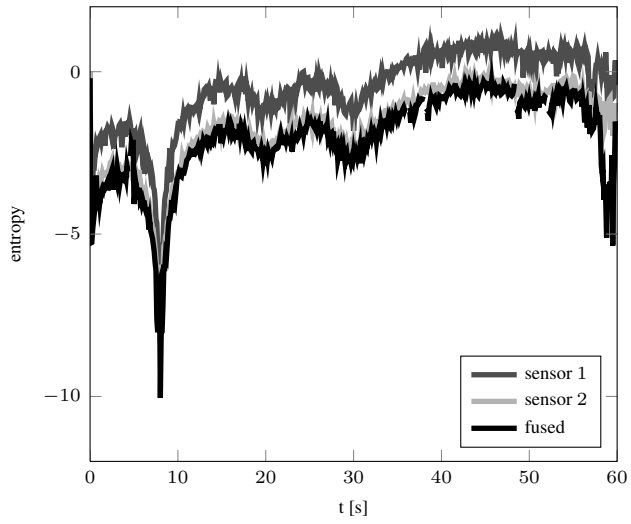

Fig. 4. Entropy of the PF *with the first sensor, with the second sensor, and the entropy of the fused hierarchical* PF *for the given synthetic example*

ing the centralized fusion, the EIF,UIF and PF had execution times of 0.20 ms, 0.95 ms and 1.84 ms, respectively, while in the case of the hierarchical fusion the aforementioned filters had execution times of 0.13 ms, 0.38 ms and 294.20 ms. In the case of the hierarchical fusion only the time of the fusion was measured, i.e. no calculation of the filter correction equations was required, just expressions (41) and (42) needed to be calculated. The reason behind the complexity of the PF fusion step is that both the predicted pdf and the posterior pdf need to be first estimated by means of kernel density estimation, thus resulting in a fusion process where the weights of the particles of the fusion filter need to be evaluated on both of the density estimate mixtures. Considering arbitrary filter fusion, for the cases when the global filter was EIF and PF the execution times were 0.38 ms and 240.28 ms, respectively.

6 CONCLUSION

In this paper we have presented Bayesian methods for sensor fusion which were divided in two groups based on the information that each sensor modality reported: a centralized independent likelihood fusion where each sensor only reported its measurement, and hierarchical fusion where each sensor ran its filter and reported its own estimate along with the uncertainty. The solution for sensor fusion in the former case was an elegant multiplication of local sensor likelihoods and the central (global) prior, while the solution for the latter case was a bit more involved and required a division of the local posterior and prior in order to extract only the new information which was then multiplied with the central (global) prior. The aforementioned approaches were given concrete expressions in the form of the EIF, UIF, and PF. The experiments were conducted on synthetic data generated from multiple Monte Carlo runs modeling a situation of dynamic object tracking with several sensor modalities.

The benefits and the importance of proper sensor fusion was demonstrated by depicting entropy of the trackers. We have shown that the fused estimates have lower entropy than the most precise sensor, even when being fused with a more imprecise sensors. Furthermore, we also discussed the problem of arbitrary sensor fusion, i.e. situations in which one sensor tracks the object with one type of a filter, e.g. a EIF, while the other tracks the object with a PF. We proposed a solution and demonstrated the approach by fusing local EIF and PF with a global PF and a global EIF. All the previous results were based on the assumption that the measurements/estimates arrive synchronously to the fusion center. Furthermore, we also discussed a solution with necessary modifications in the case of asynchronous fusion, which mostly pertained to the correct system prediction, i.e. calculation of the process noise covariance matrix.

APPENDIX A

The dynamics of the simulated moving object were governed by the nearly coordinated turn model given by

the follwing expression [20]:

$$\begin{aligned}\boldsymbol{x}_k &= \mathbf{A}\boldsymbol{x}_{k-1} + \mathbf{G}\boldsymbol{v}_k \\ &= \begin{bmatrix} 1 & \frac{\sin\omega\Delta T}{\omega} & 0 & -\frac{1-\cos\omega\Delta T}{\omega} & 0 \\ 0 & \cos\omega\Delta T & 0 & -\sin\omega\Delta T & 0 \\ 0 & \frac{1-\cos\omega\Delta T}{\omega} & 1 & \frac{\sin\omega\Delta T}{\omega} & 0 \\ 0 & \sin\omega\Delta T & 0 & \cos\omega\Delta T & 1 \end{bmatrix} \boldsymbol{x}_{k-1} \\ &+ \begin{bmatrix} \frac{\Delta T^2}{2} & 0 & 0 \\ \Delta T & 0 & 0 \\ 0 & \frac{\Delta T^2}{2} & 0 \\ 0 & \Delta T & 0 \\ 0 & 0 & \Delta T \end{bmatrix} \boldsymbol{v}_k,\end{aligned} \tag{55}$$

where the state vector from the Section 2.1 was augmented with the turn rate $\omega = 0.5$ rad/s, and the process noise was simulated with $\boldsymbol{v}_k \sim \mathcal{N}_{3\times 1}(0, 0.5)$. This solution is presented here for completeness only, and was not used in the simulations.

ACKNOWLEDGMENT

This work has been supported by European Community's Seventh Framework Programme under grant agreement no. 285939 (ACROSS).

REFERENCES

[1] J. Richardson and K. Marsh, "Fusion of Multisensor Data," *The International Journal of Robotics Research*, vol. 7, no. 6, pp. 78–96, 1988.

[2] G. A. Watson and W. Blair, "Tracking Maneuvering Targets with Multiple Sensors Using the Interacting Multiple Model Algorithm," in *Signal and Data Processing of Small Targets 1993, Proc SPIE*, vol. 1954, pp. 438–447, 1993.

[3] R. Luo and M. Kay, "Multisensor Integration and Fusion in Intelligent Systems," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 19, no. 5, pp. 901–931, 1989.

[4] D. L. Hall and J. Llinas, "An Introduction to Multisensor Data Fusion," *Proceedings of the IEEE*, vol. 85, no. 1, pp. 6–23, 1997.

[5] Y. Nakamura and Y. Xu, "Geometrical Fusion Method for Multi-Sensor Robotic Systems," in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 668–673, 1989.

[6] R. Luo, M.-H. Lin, and R. Scherp, "Dynamic Multi-Sensor Data Fusion System for Intelligent Robots," *IEEE Journal of Robotics and Automation*, vol. 4, no. 4, pp. 386–396, 1988.

[7] B. Rao, H. Durrant-Whyte, and J. Sheen, "A Fully Decentralized Multi-Sensor System For Tracking and Surveillance," *The International Journal of Robotics Research*, vol. 12, pp. 20–44, Feb. 1993.

[8] E. Nettleton, H. Durrant-Whyte, and S. Sukkarieh, "A Robust Architecture for Decentralised Data Fusion," in *The 11th International Conference on Advanced Robotics (ICAR'03)*, 2003.

[9] D.-J. Lee, "Nonlinear Estimation and Multiple Sensor Fusion Using Unscented Information Filtering," *IEEE Signal Processing Letters*, vol. 15, pp. 861–864, 2008.

[10] C. Martin, E. Schaffernicht, A. Scheidig, and H.-M. Gross, "Multi-Modal Sensor Fusion using a Probabilistic Aggregation Scheme for People Detection and Tracking," *Robotics and Autonomous Systems*, vol. 54, pp. 721–728, Sept. 2006.

[11] S. Julier and J. Uhlmann, "A Non-Divergent Estimation Algorithm in the Presence of Unknown Correlations," in *Proceedings of the 1997 American Control Conference*, pp. 2369–2373, American Autom. Control Council, 1997.

[12] M. Coates, "Distributed Particle Filters for Sensor Networks," in *Information Processing in Sensor Networks*, pp. 99–107, Springer, 2004.

[13] M. Rosencrantz, G. Gordon, and S. Thrun, "Decentralized Sensor Fusion with Distributed Particle Filters," in *Proceedings of the Conference on Uncertainty in AI (UAI)*, (Acapulco, Mexico), 2003.

[14] Y. Chen and Y. Rui, "Real-time Speaker Tracking Using Particle Filter Sensor Fusion," *Proceedings of the IEEE*, vol. 92, no. 3, pp. 485–494, 2004.

[15] L.-L. Ong, B. Upcroft, M. Ridley, T. Bailey, S. Sukkarieh, and H. Durrant-Whyte, "Consistent Methods for Decentralised Data Fusion using Particle Filters," in *2006 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, no. d, pp. 85–91, IEEE, Sept. 2006.

[16] L.-L. Ong, T. Bailey, H. Durrant-Whyte, and B. Upcroft, "Decentralised Particle Filtering for Multiple Target Tracking in Wireless Sensor Networks," in *11th International Conference on Information Fusion*, pp. 1–8, IEEE, 2008.

[17] L.-L. S. Ong, *Non-Gaussian Representations for Decentralised Bayesian Estimation*. PhD thesis, The University of Sydney, 2007.

[18] H. F. Durrant-Whyte and T. C. Henderson, "Multisensor Data Fusion," in *Springer Handbook of Robotics*, pp. 585–610, 2008.

[19] S. Jurić-Kavelj, I. Marković, and I. Petrović, "People Tracking with Heterogeneous Sensors using JPDAF with Entropy Based Track Management," in *Proceedings of the 5th European Conference on Mobile Robots (ECMR2011)*, pp. 31–36, 2011.

[20] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with Applications To Tracking and Navigation*, vol. 9. John Wiley & Sons, Inc., 2001.

[21] H. Hashemipour, S. Roy, and A. Laub, "Decentrailzed Structures for Paraller Kalman Filtering," *IEEE Transactions on Automatic Control*, vol. 33, no. 1, pp. 88–94, 1988.

[22] S. J. Julier and J. K. Uhlmann, "A New Extension of the Kalman Filter to Nonlinear Systems," in *International Symposium on Aerospace/Defense Sensing, Simulate and Control*, vol. 54, (Orlando, FL), 1997.

[23] S. J. Julier, J. K. Uhlmann, and H. F. Durrant-Whyte, "A New Method for the Nonlinear Transformation of Means and Covariances in Filters and Estimators," *IEEE Transactions on Automatic Control*, vol. 45, no. 3, pp. 477–482, 2000.

[24] S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A Tutorial on Particle Filters for On-line Non-linear/Non-Gaussian Bayesian Tracking," *Signal Processing*, vol. 50, pp. 174–188, 2001.

[25] A. Doucet, N. de Freitas, and N. Gordon, "An Introduction to Sequential Monte Carlo Methods," in *Sequential Monte Carlo Methods in Practice* (A. Doucet, N. de Freitas, and N. Gordon, eds.), ch. 1, pp. 3–14, Springer-Verlag, 2001.

[26] D. Fox, "Adapting the Sample Size in Particle Filter Through KLD-Sampling," *International Journal of Robotics Research*, vol. 22, 2003.

[27] E. Parzen, "On Estimation of a Probability Density Function and Mode," *The Annals of Mathematical Statistics*, vol. 33, no. 3, pp. 1065–1076, 1962.

[28] C. Musso, N. Oudjane, and F. Le Gland, "Improving Regularised Particle Filters," in *Sequential Monte Carlo Methods in Practice* (A. Doucet, N. de Freitas, and N. Gordon, eds.), pp. 247–272, Springer-Verlag, 2001.

[29] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. John Wiley & Sons, Inc., 2006.

[30] K. Torkkola, "Feature Extraction by Non-Parametric Mutual Information Maximization," *Journal of Machine Learning Research*, vol. 3, pp. 1415–1438, Oct. 2003.

Ivan Marković received the B.Sc. and the Ph.D. degree in Electrical Engineering from the University of Zagreb, Faculty of Electrical Engineering and Computing (FER), Croatia in 2008 and 2014, respectively. He is currently employed at the FER, Zagreb, as a Research and Teaching Assistant funded by the Ministry of Science, Education and Sport, Republic of Croatia. During his undergraduate studies, for outstanding academic achievements, he received the "Institute for Nuclear Technology Award" and the "Josip Lončar Award" in 2007 and 2006, respectively. In 2013 and 2014 he was a visiting researcher at Inria Rennes-Bretagne Atlantique in the Lagadic group directed by Prof. François Chaumette, Ph.D. His main research interests are in mobile robotics, estimation theory and human–robot interaction.

Ivan Petrović received B.Sc. degree in 1983, M.Sc. degree in 1989 and Ph.D. degree in 1998, all in Electrical Engineering from the Faculty of Electrical Engineering and Computing (FER Zagreb), University of Zagreb, Croatia. He had been employed as an R&D engineer at the Institute of Electrical Engineering of the Končar Corporation in Zagreb from 1985 to 1994. Since 1994 he has been with FER Zagreb, where he is currently a full professor and the head of the Department of Control and Computer Engineering. He teaches a number of undergraduate and graduate courses in the field of control systems and mobile robotics. His research interests include various advanced control strategies and their applications to control of complex systems and mobile robots navigation. Results of his research effort have been implemented in several industrial products. He is a member of IEEE, IFAC – TC on Robotics and FIRA – Executive committee. He is a collaborating member of the Croatian Academy of Engineering.

AUTHORS' ADDRESSES

Ivan Marković, Ph.D.
Prof. Ivan Petrović, Ph.D.
University of Zagreb,
Faculty of Electrical Engineering and Computing,
Department of Control and Computer Engineering,
Unska 3, HR-10000, Zagreb, Croatia
email: {ivan.markovic, ivan.petrovic}@fer.hr

Received: 2014-04-24
Accepted: 2014-09-24