People Tracking with Heterogeneous Sensors using JPDAF with Entropy Based Track Management

Srećko Jurić-Kavelj Ivan Marković Ivan Petrović
Department of Control and Computer Engineering, University of Zagreb
Faculty of Electrical Engineering and Computing, Croatia

Abstract— **In this paper we study the problem of tracking an arbitrary number of people with multiple heterogeneous sensors. To solve the problem, we start with a Bayesian derivation of the multiple-hypothesis tracking (MHT), and, under certain assumptions, we arrive to the joint probabilistic data association filter (JPDAF). In their original derivation, both the MHT and JPDAF assume a multiple sensor scenario which enables us to fuse the sensors measurements by asynchronously updating the tracking filters. To solve the data association problem, instead of using the optimal MHT with complex hypothesis branching, we choose the JPDAF since we are interested only in local observations by a mobile robot for people detection, tracking, and avoidance. However, the JPDAF assumes a constant and known number of objects in the scene, and therefore, we propose to extend it with an entropy based track management scheme. The benefits of the proposed approach are that all the required data come from a running filter, and that it can be readily utilized for an arbitrary type of filter, as long as such a strong mathematical principle like entropy is tractable for the underlying distribution. The proposed algorithm is implemented for the Kalman and particle filter, and the performance is verified by simulation and experiment. For the simulation purposes, we analyze two generic sensors, a location and a bearing sensor, while in the experiments we use a laser range scanner, a microphone array and an RGB-D camera.**

Index Terms— **Multi-sensor fusion, 3D sensing, JPDAF, Entropy**

I. 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 should never reduce the performance of the optimal estimator [24]. However, in order to ensure this performance, special care must be taken when choosing the process model [29]. Furthermore, system reliability increases with additional sensors, since the system itself becomes more resilient to sensor failure [14]. Therefore, by combining data from multiple sensors, and perhaps related information from associated databases, we can achieve improved accuracies and more specific inferences than by using only a single sensor [9]. With such approach, we increase the chances of a mobile robot to operate intelligently in a dynamic environment.

Sensor measurements may be combined, or fused, at a variety of levels; from the raw data level to the state vector level, or at the decision level [9]. 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 a decision level. Some sensors, like monocular cameras and microphone arrays, can only measure the angle and not the range of the detected objects. Moreover, some sensors can provide measurements at higher rates, thus making sensor fusion an even more challenging problem.

A large body of work exists on tracking moving objects with mobile robots. As discussed in [21] two major approaches can be identified, both defined by the sensors. The first approach stems form the field of computer vision and implies a camera as a major sensor, while the second utilizes laser range sensor (LRS) whose measurements are similar to those of radars and sonars. Since the field of tracking and surveillance (where radars and sonars are commonly used), was well established before the mobile robotics, a lot of results [8], [22] from that field were applied to the problem of people tracking with an LRS. The LRS approach can be further subdivided according to data association techniques into deterministic and probabilistic [1], [11], [12], [25] approaches. Additionally, these two sensors can also be used conjointly. E.g., in [4], the nearest neighbour approach and unscented Kalman filter are used for tracking people with a laser and a camera, while in [16] the authors used euclidean distance and covariance intersection method for fusing laser, sonar and camera measurements.

When considering multitarget tracking, data association is the fundamental problem. A detailed overview of probabilistic data association techniques is given in [6]. Our previous work [11] was heavily influenced by [1], [25], where the authors use the joint probabilistic data association filter (JPDAF) to solve the data association problem. In [19] the JPDAF is extended to handle multiple data sources (sensors). Such a rigorous approach is questioned when looking at the JPDAF seminal paper [8, Section V, Fig. 2], since the target-sensor geometry indicates that three sonar sensors were used to obtain the measurements. Since in our case the data acquisition happens asynchronously across sensors, we prefer the approach in [8]. The idea is as follows. When the new sensory inputs arrive, predictions about track states are made, and then the JPDAF is used to solve the data association problem. Finally, the track states are updated according to the association probabilities, where the final steps use the likelihood function of the reporting sensor, and that is the only thing required by the JPDAF to handle the multisensor case.

Another approach to probabilistic data association is the multiple hypothesis tracking (MHT) developed in a seminal paper [22] by Reid. It is an optimal solution to the data association problem, unlike the JPDAF. As discussed by Reid himself [22, Section I], the JPDAF is a special case of the MHT, in which only one hypothesis remains after data processing. To be clear, Reid is referring to [3], which is the initial derivation of the JPDAF.

Reid already solves the multisensor problem for two different generic types of sensors. He accomplishes that by describing sensors with their detection and false-alarm statistics. Thanks to such approach, we can use any type of a sensor, provided we have its probabilistic description. The downside of the MHT is in its high memory and processing requirements (which grow exponentially with the number of tracks). However, an efficient implementation of the MHT is discussed in [7] and some recent applications are presented in [2], [13].

Instead of using the optimal MHT with complex hypothesis branching, we choose the simpler, although not optimal JPDAF, as it is a very convenient solution for people tracking by a mobile robot for its local navigation [26]. However, the JPDAF assumes a constant and known number of objects in the scene, and therefore we propose to extend it with an entropy based track management algorithm. The proposed approach is tested for, but not limited to, the Kalman and particle filter. Furthermore, both trackers were tested in simulation and experiment.

II. Problem statement

We consider initialized tracks at time k described by a set of continuous random variables $\mathbf{X}^k = \left\{\hat{\mathbf{x}}_1^k, \hat{\mathbf{x}}_2^k, \dots, \hat{\mathbf{x}}_{T_k}^k\right\}$, where T_k denotes the number of tracks. At time k we receive a set $\mathbf{Z}_{s_k} = \left\{\mathbf{z}_1^{s_k}, \mathbf{z}_2^{s_k}, \dots, \mathbf{z}_{m_k}^{s_k}\right\}$ of measurements from sensor s_k, where m_k denotes the number of measurements. A set of all measurements received until and including k is denoted as $\mathbf{Z}^k = \{\mathbf{Z}_{s_0}, \dots, \mathbf{Z}_{s_k}\}$.

Let Θ^{k-1} denote a set of hypotheses about measurement to track assignments at time $k-1$. Considering a specific hypothesis $\Theta_{p(h)}^{k-1}$ at time $k-1$, we can construct legal assignment $\theta_h(k)$ for the set of measurements $\mathbf{Z}_{s_k}$. The resulting hypothesis at time k is denoted as $\Theta_h^k = \{\Theta_{p(h)}^{k-1}, \theta_h(k)\}$, where $\Theta_{p(h)}^{k-1}$ denotes the parent of the Θ_h^k hypothesis.

We can now consider the probability of a hypothesis given the sensors measurements $P\left(\Theta_h^k \middle| \mathbf{Z}^k\right)$ and calculate it using Bayes' theorem

$$\begin{aligned} P\left(\Theta_h^k \middle| \mathbf{Z}^k\right) &= \frac{P\left(\Theta_{p(h)}^{k-1}, \theta_h(k), \mathbf{Z}_{s_k}, \mathbf{Z}^{k-1}\right)}{P\left(\mathbf{Z}_{s_k}, \mathbf{Z}^{k-1}\right)} \\ &= P\left(\mathbf{Z}_{s_k} \middle| \Theta_{p(h)}^{k-1}, \theta_h(k), \mathbf{Z}^{k-1}\right) \\ &\quad \cdot P\left(\theta_h(k) \middle| \Theta_{p(h)}^{k-1}, \mathbf{Z}^{k-1}\right) \\ &\quad \cdot \frac{P\left(\Theta_{p(h)}^{k-1} \middle| \mathbf{Z}^{k-1}\right)}{P\left(\mathbf{Z}_{s_k} \middle| \mathbf{Z}^{k-1}\right)} . \end{aligned} \tag{1}$$

The JPDAF can be viewed as a special case of the MHT, where after each iteration all the considered hypotheses are reduced to a single hypothesis. We denote that hypothesis at time k as $\theta(k)$. Again, the JPDAF considers a possible data assignment hypothesis relative to the hypothesis in the previous time instant $k-1$. Calculation of the probability for such hypothesis reduces to

$$\begin{aligned} P\left(\Theta_h^k \middle| \mathbf{Z}^k\right) = \frac{1}{c} & P\left(\mathbf{Z}_{s_k} \middle| \theta_h(k), \theta(k-1), \mathbf{Z}^{k-1}\right) \\ & \cdot P\left(\theta_h(k) \middle| \theta(k-1), \mathbf{Z}^{k-1}\right) \\ & \cdot P\left(\theta(k-1) \middle| \mathbf{Z}^{k-1}\right) , \end{aligned} \tag{2}$$

where c is a normalizing constant. Since we stated that the JPDAF is a zero scan data association algorithm (only one hypothesis is left after the measurements processing), the probability of the previous hypothesis $P\left(\theta(k-1) \middle| \mathbf{Z}^{k-1}\right)$ is equal to one. If no new track hypotheses are considered, then this formulation coincides with the one in [8].

Each hypothesis Θ_h^k contains track states $\mathbf{X}^k$ updated using considered association $\theta_h(k)$. Since the JPDAF *flattens* the hypothesis tree to a single branch $\theta(k)$, it contains track states $\mathbf{X}^k$ updated using all the measurements, given their association probabilities

$$\beta_j^t = \sum_{\theta \in \Theta_{jt}^k} P\left(\theta \middle| \mathbf{Z}^k\right) , \tag{3}$$

where Θ_{jt}^k denotes all the hypotheses that associate measurement j with track t.

III. Joint Probabilistic Data Association Filter

As stated before, the JPDAF considers possible data assignment hypothesis $\theta_h(k)$ relative to the singular hypothesis from the previous time instant $k-1$, which has unit probability. Probability of a specific hypothesis is given in (2) (with $P\left(\theta(k-1) \middle| \mathbf{Z}^{k-1}\right) = 1$). This leaves us to describe the other two terms in (2). The second term, $P\left(\theta_h(k) \middle| \theta(k-1), \mathbf{Z}^{k-1}\right)$ can be modeled as a constant, as in [11], [25]. A more precise derivation of this term can be found in [7], [8], [22]. Now, we only need to develop the first term

$$\begin{aligned} P\left(\mathbf{Z}_{s_k} \middle| \theta_h(k), \theta(k-1), \mathbf{Z}^{k-1}\right) &= P\left(\mathbf{Z}_{s_k} \middle| \theta_h(k)\right) \\ &= \prod_{j=1}^{m_k} P\left(\mathbf{z}_j^{s_k} \middle| \theta_h(k)\right) , \end{aligned} \tag{4}$$

where $P\left(\mathbf{z}_j^{s_k} \middle| \theta_h(k)\right)$ depends on the measurement to track associations made by hypothesis $\theta_h(k)$

$$P\left(\mathbf{z}_j^{s_k} \middle| \theta_h(k)\right) = \begin{cases} P_F^{s_k}, & \mathbf{z}_j^{s_k} \text{ false alarm} \\ P_D^{s_k} P\left(\mathbf{z}_j^{s_k} \middle| \hat{\mathbf{x}}_t^k\right), & \mathbf{z}_j^{s_k} \text{ existing track} \end{cases} \tag{5}$$

where $P_D^{s_k}$ is the detection and $P_F^{s_k}$ the false alarm probability for sensor s_k.

By inserting (5) in (2) and then inserting the result in (3), we get an expression for measurements to tracked objects association probabilities

$$\beta_j^t = \frac{1}{c} \sum_{\theta \in \Theta_{jt}^k} \prod_{j=1}^{m_k} P\left(\mathbf{z}_j^{s_k} \middle| \theta\right) . \tag{6}$$

Aforementioned association probabilities are used to update the track states (the filtering part). Since the track states are updated with all the measurements (weighted by their respective probabilities) from their cluster, this is what essentially flattens the hypothesis tree. In other words, after the update, only one hypothesis remains, the one about the current track states.

The actual track state update and implementation of (6) depends on the used state estimator (filter). In this paper, we present a particle filter, used in our previous work [11], and a Kalman filter, used in the original JPDAF formulation [8].

A. Kalman JPDAF

Given any state estimator, a process model is required. We use a quite general constant velocity model for motion in 2D plane, where state

$$\mathbf{x} = [x\ \dot{x}\ y\ \dot{y}]^T \tag{7}$$

is described by position (x, y) and velocity $(\dot{x}, \dot{y})$ in the xy–plane. The model itself is given by

$$\begin{aligned}\mathbf{x}^{k+1} &= \mathbf{F}\mathbf{x}^k + \mathbf{G}w_k \\ &= \begin{bmatrix}1 & \Delta T_k & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 1 & \Delta T_k\\ 0 & 0 & 0 & 1\end{bmatrix}\mathbf{x}^k + \begin{bmatrix}\frac{\Delta T_k^2}{2} & 0\\ \Delta T_k & 0\\ 0 & \frac{\Delta T_k^2}{2}\\ 0 & \Delta T_k\end{bmatrix} w_k,\end{aligned} \tag{8}$$

where w_k is the process noise and ΔT_k is the update interval.

Prediction is calculated using standard Kalman filter equations

$$\begin{aligned}\hat{\mathbf{x}}_t^{k-} &= \mathbf{F}\hat{\mathbf{x}}_t^{k-1}\ ,\\ \mathbf{P}_t^{k-} &= \mathbf{F}\mathbf{P}_t^{k-1}\mathbf{F}^T + \mathbf{G}\mathbf{Q}\mathbf{G}^T\ .\end{aligned} \tag{9}$$

Given the innovation vector

$$\nu_j^t = \mathbf{z}_j^{s_k} - \mathbf{H}\hat{\mathbf{x}}_t^{k-}, \tag{10}$$

and its covariance matrix

$$\mathbf{S}_t = \mathbf{H}\mathbf{P}_t^{k-}\mathbf{H}^T + \mathbf{R}_{s_k}, \tag{11}$$

we can define (5) in the case of an existing track association as $P\left(\mathbf{z}_j^{s_k}\left|\mathbf{x}_t^k\right.\right) = \mathcal{N}(\nu_j^t; 0, \mathbf{S}_t)$.

The innovation vector and covariance matrix can be used for measurement gating. Since ${\nu_j^t}^T\mathbf{S}_t^{-1}\nu_j^t$ has χ^2 distribution, by using tables we can select upper limit which includes valid measurements with, e.g., 99% probability.

Update is done by using all the validated measurements, i.e. weighted innovation is used for the state update

$$\begin{aligned}\nu^t &= \sum_{j=1}^{m_k}\beta_j^t\nu_j^t\ ,\\ \hat{\mathbf{x}}_t^k &= \hat{\mathbf{x}}_t^{k-} + \mathbf{K}_k\nu^t\ .\end{aligned} \tag{12}$$

Given $\beta^t = 1 - \sum_{j=1}^{m_k}\beta_j^t$ and $\mathbf{P}_{\nu^t} = \sum_{j=1}^{m_k}\beta_j^t\nu_j^t{\nu_j^t}^T - \nu^t{\nu^t}^T$ the covariance update is calculated as in [5]

$$\mathbf{P}_t^k = \beta^t\mathbf{P}_t^{k-} + (1-\beta^t)\left[\mathbf{I} - \mathbf{K}_k\mathbf{H}\right]\mathbf{P}_t^{k-} + \mathbf{K}_k\mathbf{P}_{\nu^t}\mathbf{K}_k^T\ . \tag{13}$$

An important implementation note is that instead of the standard Kalman filter covariance update $\left[\mathbf{I} - \mathbf{K}_k\mathbf{H}\right]\mathbf{P}_t^{k-}$ we use Joseph's stabilized form $\left[\mathbf{I} - \mathbf{K}_k\mathbf{H}\right]\mathbf{P}_t^{k-}\left[\mathbf{I} - \mathbf{K}_k\mathbf{H}\right]^T + \mathbf{K}_k\mathbf{R}_{s_k}\mathbf{K}_k^T$, since the standard form caused numerical problems.

B. Particle JPDAF

If a particle filter is to be used as a state estimator, besides the prediction and update of the filter, we also have to modify gating and hypotheses probability calculation procedures. We model the estimated state as a set of tuples $(\hat{\mathbf{x}}_t^k(i), w_t(i))$ where $\hat{\mathbf{x}}_t^k(i)$ is a particle containing a possible track state, and $w_t(i)$ is its associated weight. If a particle falls within the valid measurement region (based on the squared Mahalanobis distance ${\nu_j^t}^T(i)\mathbf{R}_{s_k}^{-1}\nu_j^t(i)$), then we consider the association of measurement j to track t when generating the hypothesis. As for the hypothesis probability calculation, we use average likelihood of the particles

$$P\left(\mathbf{z}_j^{s_k}\left|\mathbf{x}_t^k\right.\right) = \frac{1}{N}\sum_{i=1}^{N}\mathcal{N}(\nu_j^t(i); 0, \mathbf{R}_{s_k}) \tag{14}$$

where $\nu_j^t(i)$ is the individual particle's innovation. After obtaining association probabilities, we update the particle weights according to $w_t(i) = \sum_{j=1}^{m_k}\beta_j^t\mathcal{N}(\nu_j^t(i); 0, \mathbf{R}_{s_k})$. After the weights calculation, and only if there are measurements in the current time step, we resample the particles.

IV. Track management

When tracking multiple targets, track management is practically as important as the association itself. A solution for the Kalman filter, described in [5], is based on a logarithmic hypothesis ratio and innovation matrix. In [25] a Bayesian estimator of the number of objects for an LRS is proposed. This approach requires learning the probability of how many features are observed under a presumed number of objects in the perceptual field of the sensor, while the tracking performance is monitored by an average of the sum of unnormalized sample weights of the particle filter.

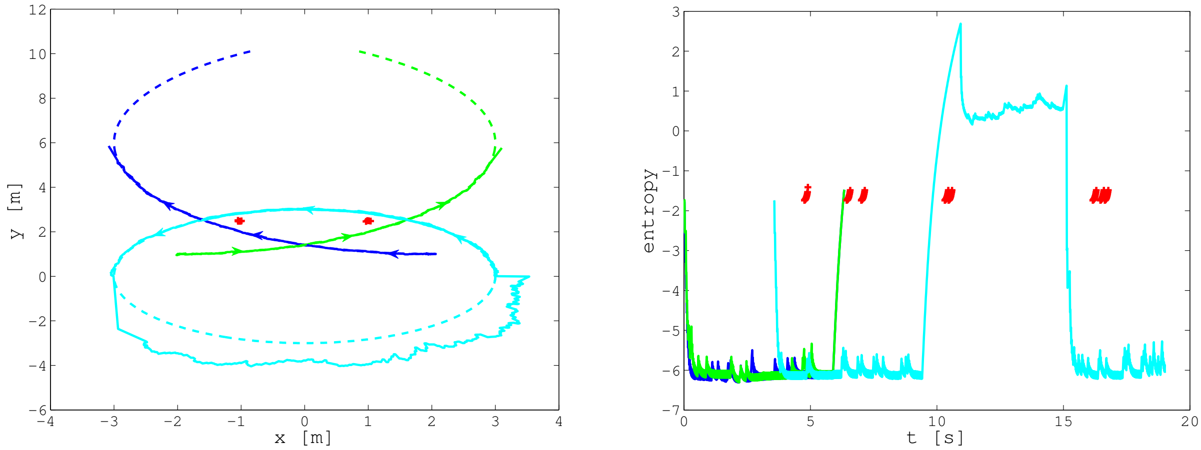
We propose to use an entropy measure as a feature in track management. If such a strong mathematical principle is tractable for the underlying probability distribution, then it can be readily utilized for track management independently of the filtering approach. Furthermore, all the information required for the entropy calculation is already available in the running filter and sensor model, and as it will be presented, threshold setting is quite convenient.

A practical entropy measure for this task is the quadratic Rényi entropy [23]

$$H_2(\hat{\mathbf{x}}_t) = -\log\int p\left(\hat{\mathbf{x}}_t\right)^2\, d\hat{\mathbf{x}}_t. \tag{15}$$

For the Kalman filter, i.e. Gaussian distributions, an analytical solution exists and is given by $H_2(\hat{\mathbf{x}}_t) = \frac{n}{2}\log 4\pi + \frac{1}{2}\log|\mathbf{P}_t|$, where n is the state dimension.

Entropy calculation of continuous random variables is based on the probability density functions (pdfs) of these variables. In order to calculate entropy of a particle filter, which rather represents the density and not the function, we need a nonparametric method to estimate the pdf. One such method is the Parzen window method [20] which involves placing a kernel function on top of each sample and then evaluating the density



(a) Trajectories

(b) KF entropies

Fig. 1. Simulation results for the KF with modeled detection probability, false alarms and silent periods – true (dashed) and estimated (solid) track states, and tentative but not confirmed tracks (red + marker). The first two objects were detected and a KF was initialized for each one at 0 s. At 5.9 s they went out the range of the sensors and the entropy of their KF kept rising until they were deleted at 6.3 s. The third object was detected at 3.6 s, and at 9.4 it went behind the robot thus causing a rise in entropy. At 11 s it was detected again by the bearing sensor causing an effective drop in entropy. At 15.1 s it moved in front of the robot again and was detected by the location sensor which significantly lowered the entropy.

as a sum of the kernels. We continue this approach as proposed in [17], [18], and convert each sample to a kernel

$$K_h(\hat{\mathbf{x}}_t) = h^n K(\hat{\mathbf{x}}_t), \tag{16}$$

where $K(.)$ is the particle set covariance, and $h > 0$ is the scaling parameter. For the kernel, we choose $h = \left(\frac{4}{n+2}\right)^e N^{-e}$, where $e = \frac{1}{n+4}$, and N is the number of particles. At this point, each track is described as a sum of Gaussian kernels, $p(\hat{\mathbf{x}}_t) = \sum_{i=1}^{N} \mathcal{N}(\hat{\mathbf{x}}_t(i), 2K_h(\hat{\mathbf{x}}_t))$, for which an analytical solution for the quadratic Rényi entropy exists [27]

$$H_2(\mathbf{x}_t) = -\log \frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} \mathcal{N}(\hat{\mathbf{x}}_t(i) - \hat{\mathbf{x}}_t(j); 0, 2K_h(\hat{\mathbf{x}}_t)). \tag{17}$$

Due to symmetry, only half of these kernels need to be evaluated in practice.

The track management logic is as follows. When the tracks are initialized, they are considered *tentative* and the initial entropy is stored. When the entropy of a tentative track drops for 50% – it is a *confirmed* track. If and when the entropy gets 20% larger than the initial entropy – the track is deleted. This logic reflect the fact that if the entropy is rising, we are becoming less and less confident that the track is informative. Furthermore, since no entropy should be greater than the one calculated at the point of the track initialization, we can use this initialization entropy as an appropriate deletion threshold.

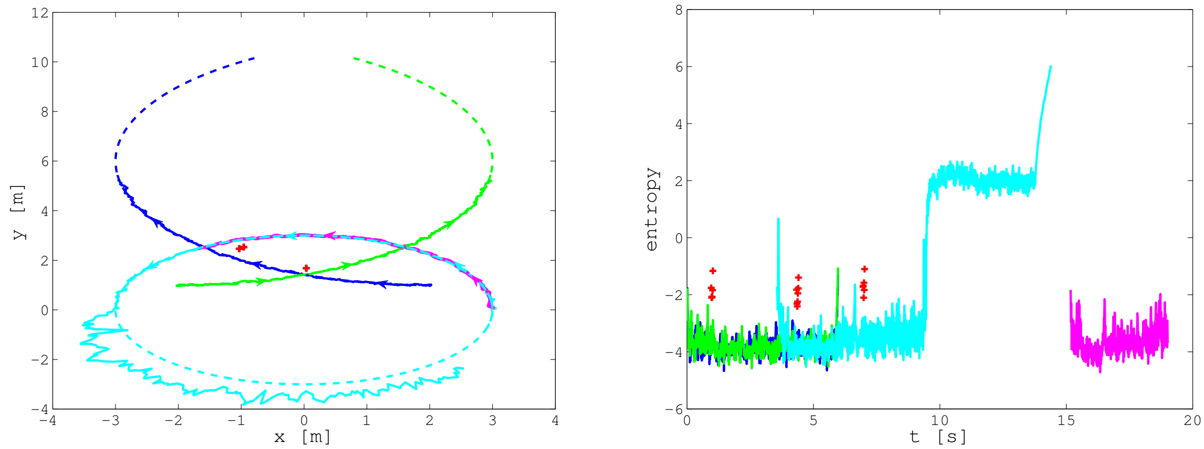
V. Simulations

In order to test the performance of the algorithm, we generated three intersecting circular trajectories. The robot was at $(0, 0, 90^\circ)$ m, the first object started at $(2, 1)$ m and finished at $(-0.8, 10)$ m, the second object started at $(-2, 1)$ m and finished at $(0.8, 10)$ m, while the third object started at $(3, 0)$ m and finished at $(-1.6, 2.5)$ thus making more than one revolution around the mobile robot (Fig. 1a). Each object was tracked in an alternating manner by the location and bearing sensor, while the maximum range for both was kept at 6 m. The location sensor can only track objects in front of the mobile robot, i.e. from 0 to π, and was corrupted with white Gaussian noise given by $\mathcal{N}([x\ y]^T; \mathbf{0}, 0.03 \cdot \mathbf{I})$. The bearing sensor, on the other hand, can only measure the bearing angle θ of the object, but in the full range around the mobile robot, i.e. from 0 to $\pm\pi$, and was also corrupted with white Gaussian noise given by $\mathcal{N}(\theta; 0, 3^\circ)$.

Furthermore, for both sensors each measurement had the detection probability of $P_D = 0.9$, and the probability of a false alarm was $P_F = 0.01$. Since the bearing sensor models a microphone array, it is logical to assume that the speaker will have pauses while talking, thus resulting in longer periods of absent measurements. This was modeled by placing a random number of pauses of maximum length of 2 seconds at random time instances. Although the assumption about the talking speaker might not be realistic for every-day scenarios, we find it important to analyze performance of a bearing-only sensor in such a multisensor system.

The tracks can only be initialized by the location sensor, but the existing tracks should be kept by the bearing sensor when the object moves behind the robot. Since in this case the entropy is substantially larger, it requires calculation of bearing-only initialization entropy, in order to efficiently manage the case when the object is behind the robot.

In Fig. 1 we show KF simulation results with added detection probability, false alarms, and silent speaker periods, from which we can see that there were several false tracks initiated but never confirmed. Furthermore, we made 100 Monte Carlo (MC) runs for the Kalman filter – on average there were 11.43 initialized, 7.95 tentative tracks, and 3.48 confirmed tracks. In an ideal situation we would have had three confirmed tracks, but taking the scenario into account we can conclude that the



(a) Trajectories (b) PF entropies

Fig. 2. Simulation results for the PF with modeled detection probability, false alarms and silent periods – true (dashed) and estimated (solid) track states, and tentative but not confirmed tracks (red + marker). The objects trajectories were the same as in the case of the KF. The main difference is in the third object's estimated trajectory: when the object moved behind the robot, there were bearing measurements up to 13.8 seconds when a silent period started. The entropy kept rising and the object was deleted before new measurements appeared. The object moved in front of the robot at 15.1 and consequently a new filter was initialized.

algorithm performs well when it comes to tracking, association and track management.

The results of the simulation for the PF with added detection probability, false alarms, and silent speaker periods are shown in Fig. 2. Furthermore, we also made 100 Monte Carlo (MC) runs for the particle filter – on average there were 8.17 initialized, 3.22 tentative, and 4.95 confirmed tracks. Although the average number of confirmed tracks was larger than in the case of the Kalman filter, we still find it to be of acceptable performance.

Simulations were performed on a machine running at 2.33 GHz with an unoptimized Matlab implementation. The average computational time of each iteration was 1.9 ms and 137.2 ms for the KF and PF, respectively. Time spent on the entropy calculation was 0.02 ms and 88.6 ms for the KF and PF, respectively.

VI. Experiments

To further test the proposed approach, we conducted experiments with our Pioneer 3-DX robot. The laser sensor was the Sick LMS 200 model, while the microphone array is of our design. Furthermore, since the proposed framework is easily extended to multiple sensors, we also used the Kinect time-of-flight camera with a face recognition algorithm based on [28] to yield a set of measurements in 3D. In the experiment two people were walking in an intersecting trajectory in front of the robot (a snapshot of the experiment is shown in Fig. 3). The results are shown in Fig. 4 from which we can see that the first person (blue line) started at $(-1.2, 2.3)$ m and finished at $(0.9, 2.3)$ m, while the second person (green line) started at $(0.7, 0)$ m and finished at $(0.6, 0)$ m. The first person was in the field-of-view (FOV) of all the three sensors and was talking throughout the experiment, while the second person entered LRS FOV at a later time, kept quiet and was facing the robot only in the second half of the trajectory. Tracks were correctly initialized and maintained, despite the large number of false alarms. The second track was deleted short-after the second person left the LRS FOV.

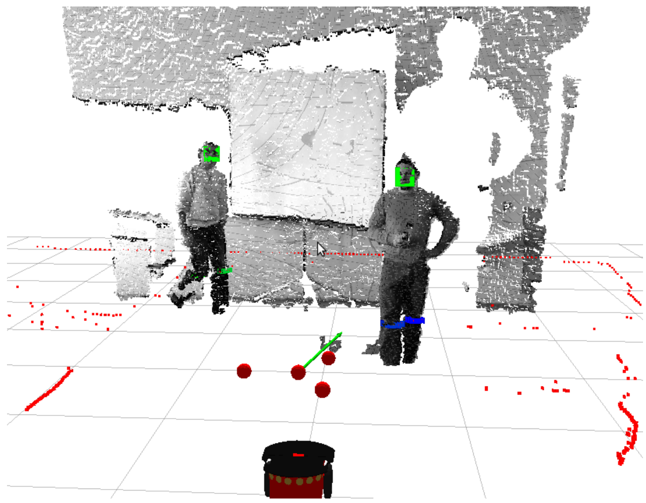
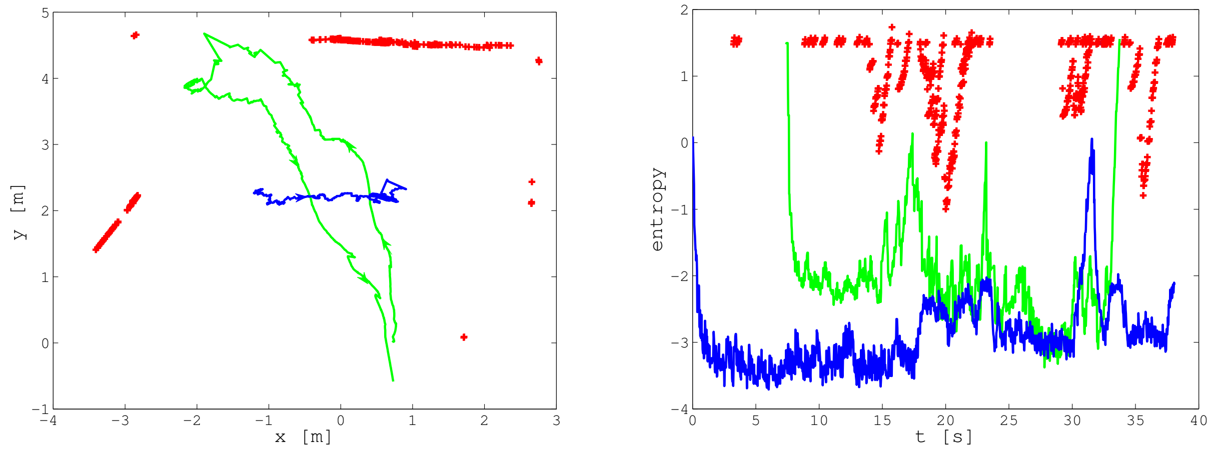


Fig. 3. A snapshot of the data acquisition and signal processing for the experiments. The measurements were classified and collected based on our previous work [10], [11], [15], with only the signal processing stage done, i.e. no tracking was performed on the sensor level.

VII. Conclusion

In the present work we addressed the problem of tracking multiple objects with multiple heterogeneous sensors – specifically an LRS, a microphone array, and an RGB-D camera. The integration of multiple sensors is solved by asynchronously updating the tracking filters as new data arrives. We solved the data association problem by applying the JPDAF, which is a suboptimal zero-scan derivation of the MHT, but which in effect assumes a known number of objects. To circumvent this assumption, we proposed an entropy based track management



(a) People trajectories

(b) KF entropies

Fig. 4. Experimental results for the KF – estimated (solid) track states, and tentative but not confirmed tracks (red + marker). The first object was in the scene from the beginning, while the second object entered the scene at 7.5 s. At 15 s the second object got occluded by the first, which caused an increase in entropy, while at 30 s the second object occluded the first shortly before exiting the scene. The false alarms were caused by tiles on the wall and leg-like features in the room (chairs and tables).

scheme, and demonstrated its performance for the Kalman and particle filter both in simulation and experiment. The results showed that the proposed algorithm is capable of maintaining a viable number of filters with correct association and accurate tracking.

ACKNOWLEDGMENT

This work was supported by the Ministry of Science, Education and Sports of the Republic of Croatia under grant No. 036-0363078-3018.

REFERENCES

[1] A. Almeida, J. Almeida, and R. Araújo. Real-time Tracking of Multiple Moving Objects Using Particle Filters and Probabilistic Data Association. *AUTOMATIKA Journal*, 46(1-2):39–48, 2005.

[2] K.O. Arras, S. Grzonka, M. Luber, and W. Burgard. Efficient people tracking in laser range data using a multi-hypothesis leg-tracker with adaptive occlusion probabilities. In *IEEE ICRA*, pages 1710–1715. IEEE, 2008.

[3] Y. Bar-Shalom. Extension of the probabilistic data association filter to multi-target environment. In *Proceedings of the 5th Symposium on Nonlinear Estimation*, pages 16–21, San Diego, CA, 1974.

[4] N. Bellotto and H. Hu. Vision and Laser Data Fusion for Tracking People with a Mobile Robot. In *Proceedings of the IEEE International Conference on Robotics and Biomimetics*, pages 7–12, 2006.

[5] S. Blackman and R. Popoli. *Design and Analysis of Modern Tracking Systems (Artech House Radar Library)*. Artech House Publishers, 1999.

[6] I.J. Cox. A review of statistical data association techniques for motion correspondence. *International Journal of CV*, 10(1):53–66, 1993.

[7] I.J. Cox and S.L. Hingorani. An Efficient Implementation of Reid's Multiple Hypothesis Tracking Algorithm and its Evaluation for the Purpose of Visual Tracking. *Trans. on PAMI*, 18(2):138–150, 1996.

[8] T. Fortmann, Y. Bar-Shalom, and M. Scheffe. Sonar tracking of multiple targets using joint probabilistic data association. *IEEE Journal of Oceanic Engineering*, 8(3):173–184, July 1983.

[9] D.L. Hall and J. Llinas. An Introduction to Multisensor Data Fusion. *Proceedings of the IEEE*, 85(1):6–23, 1997.

[10] S. Jurić-Kavelj and I. Petrović. Experimental comparison of AdaBoost algorithms applied on leg detection with different range sensor setups. In *19th International Workshop on RAAD*, pages 267–272. IEEE, 2010.

[11] S. Jurić-Kavelj, M. Seder, and I. Petrović. Tracking Multiple Moving Objects Using Adaptive Sample-based Joint Probabilistic Data Association Filter. In *Proceedings of 5th International Conference on CIRAS*, pages 99–104, Linz, Austria, 2008.

[12] A. Kräußling and D. Schulz. Tracking extended targets—a switching algorithm versus the SJPDAF. In *Proc. of FUSION*, pages 1–8, Florence, Italy, 2006.

[13] M. Luber, G.D. Tipaldi, and K.O. Arras. Spatially grounded multi-hypothesis tracking of people. In *Workshop on People Detection and Tracking, 2009 IEEE ICRA*, Kobe, Japan, 2009.

[14] R. Luo and M. Kay. Multisensor Integration and Fusion in Intelligent Systems. *IEEE Transactions on Systems, Man, and Cybernetics*, 19(5):901–931, 1989.

[15] I. Marković and I. Petrović. Speaker Localization and Tracking with a Microphone Array on a Mobile Robot using von Mises Distribution and Particle Filtering. *Robotics and Autonomous Systems*, 58(11):1185–1196, November 2010.

[16] 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*, 54(9):721–728, September 2006.

[17] C. Musso, N. Oudjane, and F. Le Gland. *Improving Regularised Particle Filters*, pages 247–272. Springer-Verlag, 2001.

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

[19] L.Y. Pao and S.D. O'Neil. Multisensor fusion algorithms for tracking. In *Proceedings of the ACC*, pages 859–863, San Francisco, CA, 1993.

[20] E. Parzen. On Estimation of a Probability Density Function and Mode. *The Annals of Mathematical Statistics*, 33(3):1065–1076, 1962.

[21] E. Prassler, J. Scholz, and A. Elfes. Tracking multiple moving objects for real-time robot navigation. *Autonomous Robots*, 8(2):105–116, 2000.

[22] D. Reid. An algorithm for tracking multiple targets. *IEEE Transactions on Automatic Control*, 24(6):843–854, December 1979.

[23] A. Rényi. *Probability Theory*. North-Holland, London, 1970.

[24] J. Richardson and K. Marsh. Fusion of Multisensor Data. *The International Journal of Robotics Research*, 7(6):78–96, 1988.

[25] D. Schulz, W. Burgard, D. Fox, and A.B. Cremers. People Tracking with Mobile Robots Using Sample-Based Joint Probabilistic Data Association Filters. *The International JRR*, 22(2):99–116, February 2003.

[26] M. Seder and I. Petrović. Dynamic window based approach to mobile robot motion control in the presence of moving obstacles. In *Proceedings of IEEE ICRA*, pages 1986–1991, Roma, Italy, 2007.

[27] K. Torkkola. Feature Extraction by Non-Parametric Mutual Information Maximization. *Journal of Machine Learning Research*, 3(7-8):1415–1438, October 2003.

[28] P. Viola and M. Jones. Robust real-time object detection. *International Journal of Computer Vision*, 2002.

[29] G.A. Watson and W.D. Blair. Tracking Maneuvering Targets with Multiple Sensors Using the Interacting Multiple Model Algorithm. *Proceedings of SPIE*, 1954:438–449, 1993.