ISSN 0005-1144
ATKAFF 50(3–4), 205–214(2009)

Dražen Brščić, Hideki Hashimoto

Sensing in Intelligent Spaces: Joint Use of Distributed and Onboard Sensors

UDK 004.896:681.586
IFAC 5.9.3;4.2.2

Original scientific paper

This work considers the joint use of robot onboard sensors and a network of sensors distributed in the environment for tracking the position of the robot and other objects. This is motivated by our research on Intelligent Spaces, which combine the use of distributed sensors with mobile robots to provide various services to users. Here we analyze the distributed sensing using the extended information filter and computation issues that arise due to correlations between estimates. In turn we show how the correlations can be resolved with the use of Covariance Intersection at a cost of conservative estimates, and analyze two special cases where the issues related to correlations can be reduced.

Key words: Intelligent spaces, Mobile robots, Tracking, Sensor network

Percepcija u inteligentnim prostorima: kombinirana primjena distribuiranih i robotskih senzora. Ovaj rad razmatra kombiniranu primjenu senzora na mobilnim robotima i mreže senzora distribuiranih u prostoru za praćenje položaja robota i ostalih objekata. Rad je dio istraživanja o "inteligentnim prostorima", gdje se koriste distribuirani senzori i mobilni roboti sa svrhom pružanja različitih usluga korisnicima prostora. Analizirana je upotreba proširenog informacijskog filtra za distribuiranu percepciju te računski problem uzrokovan korelacijama u procesu estimacije. Potom je objašnjeno rješenje problema korelacija korištenjem metode presjeka kovarijanci (Covariance Intersection), koje međutim daje konzervativne rezultate, te je dana analiza dva specijalna slučaja kod kojih je moguće ublažiti utjecaj korelacija.

Ključne riječi: inteligentni prostori, mobilni roboti, praćenje, mreža senzora

1 INTRODUCTION

The research area of ambient intelligence [1–4], which investigates spaces with distributed sensing devices used to detect human users in order to provide services in accordance with their needs, has been expanding. Furthermore, there is a growing number on research works on environments with ubiquitously distributed sensors where robotic technology is also included as a mean of providing physical actions [5–9]. While there are several names that are often used for such environments, we refer to them here as "Intelligent Spaces" or iSpaces. Intelligent Spaces can therefore be considered a combination of ambient intelligence and robotics, designed with the purpose of supporting human users in both informative and physical ways [5], Fig. 1. They rely on a network of sensing devices to obtain information about the space and its state, whereas multiple actuators provide services to the users. These include both informative services, for example provided using various displays or projectors, and physical services that can be provided using actuated devices or mobile robots.

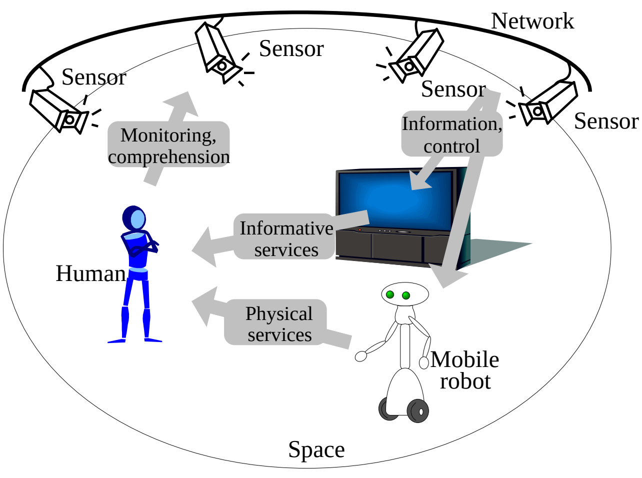

Fig. 1. Concept of Intelligent Spaces

One essential function required by an Intelligent Space is sensing, or more precisely the ability to track the position of objects in the space – most importantly that of humans and mobile robots. Two distinct approaches to sensing can be applied: in the research works on ambient intelligence tracking is mostly done using a network of distributed sensors, whereas in robotics it is typical to use the robot's onboard sensors. Since both types of sensors exist in an iSpace, it is beneficial to combine their use, which is the topic of this work.

A variety of sensors have been utilized for tracking in iSpaces, with the most common one being networks of cameras distributed in the space, see e.g. [2, 9]. On the other hand, most commonly used sensors for localization and tracking in mobile robotics nowadays are laser range finders. In the implementation in this work we use laser range finders both as distributed and onboard sensors. Note, however, that most of the results are general and can be applied to any type of sensor.

In [10] it was argued that a flexible implementation of Intelligent Spaces requires a hierarchical architecture, and a four layer structure consisting of sensor, information server, applications and actuator layers was proposed (Fig. 2). The connections exist only between adjacent layers: information servers fuse the information from sensors and provide it to applications, which in turn send the commands to the actuators to realize various services. This structure makes it easy to add or change parts of the architecture without affecting the rest of the system thereby providing high flexibility and modularity.

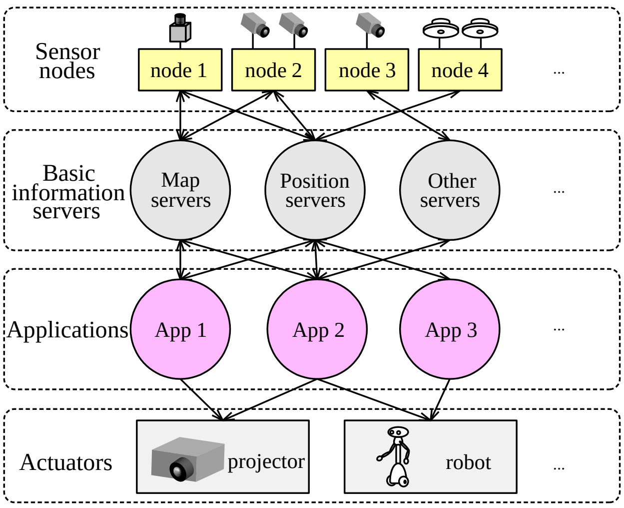

Fig. 2. Hierarchical implementation architecture of Intelligent Space

This results in a centralized architecture for sensing and sensor fusion, such as the one shown in Fig. 3. All the sensors connect through their local nodes or through the robot to an information server (the fusion server), which serves as the central processing unit.

In this work we analyze the problem of tracking of robots and other objects using both distributed and onboard sensors. The starting point is the just described centralized architecture and a distributed estimation algorithm based on the extended information filter. Section 3 describes the estimation method and the issue of correlations between estimates due to the use of onboard sensors. Ways to deal with the correlation problem are described in sections 4 and 5.

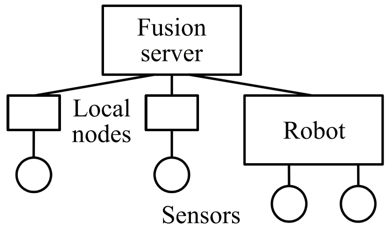

Fig. 3. Centralized estimation

This paper focuses only on estimation methods that use Gaussian representations of state and noises – an example of non-Gaussian estimation can be found in [11].

2 RELATED WORK

Different implementations of tracking of humans and robots in the ambient intelligence and iSpace research can be found in literature. For example in [7] humans wearing RFID tags were tracked using a system of RFID readers, while infrared cameras were used for tracking robots. In [12] RFID tags are used for robot localization as landmarks of absolute position. In another work [13] the authors use vision based artificial landmarks to track the robot. In these works robot's onboard sensors were not used (apart from encoders).

There has been a large amount of research work on laser range finder based tracking/localization, especially in the mobile robotics community. This work is mainly concerned with the estimation of the robot's own pose, i.e. robot localization, while scanning the surrounding area using a laser range finder mounted on the robot. A very large part of this work is concerned with the so called simultaneous localization and mapping (SLAM) problem, where the robot tries to simultaneously estimate its own location and build a map of the environment. There is also some work concerned with the detection and tracking of humans in the vicinity of robots [14, 15].

There are several papers dealing with the problem of tracking objects when using laser range finders distributed at fixed locations in the space. In [16] laser range finders are used for tracking people in everyday environments, whereas in [17] the position of pedestrians in large open spaces were estimated.

In [18] tracking using laser range finders was applied to extract trajectories of persons, which are further used to learn human motion patterns. They did experiments both with distributed and with onboard laser range finders. Similarly, in [19] the authors tried to detect anomalies in human interaction based on the position tracking with laser range finders.

In [20] a strategy for deployment of robots as mobile sensors in an environment equipped with distributed and networked sensors was developed with the purpose of improving the tracking of targets in the environment. A recent work in [21] proposes a active strategy for global localization of mobile robots equipped with onboard sensors in an environment with distributed sensors. In both these works the main concern is how to plan the robot movement and not how to perform the tracking or localization.

There are only a few works where the information from both onboard and distributed sensors was combined to obtain a better estimate. One such example can be found in [6]. Here the humans were tracked using an ultrasound system, while the robot used self-localization based on onboard laser range finder readings. Since humans can be detected by the onboard sensor, this information was fused with the ultrasound system reading to obtain a better estimate.

3 SENSING IN INTELLIGENT SPACES

3.1 System description

The system setup considered in this work is depicted in Fig. 4. It consists of several sensors (laser range finders) distributed at fixed locations in the space, and mobile robots that are equipped with onboard laser range finders. It is assumed that both distributed and onboard sensors can observe other objects in the space – either humans or static landmarks. Distributed sensors can in addition detect the mobile robot.

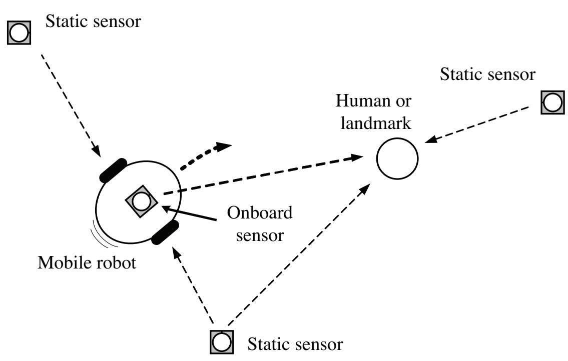

Fig. 4. Tracking system with both distributed and onboard sensors

Typical problems that are considered when using distributed and onboard sensors are robot tracking, human tracking and mapping of the space. Accordingly, the states that are estimated can be the robot pose $\mathbf{x}^r = [\ x^r \quad y^r \quad \theta^r\ ]^T$, position of the humans if human tracking is included: $\mathbf{x}^h = [\ x^h \quad y^h\ ]^T$, or position of static landmarks in the space $\mathbf{x}^l = [\ x^l \quad y^l\ ]^T$.

Table 1. Comparison between sensing with distributed and with onboard sensors.

	Robot	Human	Map
Distributed	Good (orientation!)	Good	Poor
Onboard	Good	Difficult	Good

Before proceeding, it is worth noting here the main characteristics of distributed or onboard sensors in the mentioned sensing problems. These are summarized in Table 1. Although both types of sensors can provide good robot position tracking / self-localization, an issue that often arises with distributed laser range finders (however not with some other sensors, like cameras) is the inability to directly detect the robot orientation. This is because a 2D laser scan of a robot typically does not have many characteristic features, thus making hard the direct estimation of orientation. It is nevertheless possible to make an estimation of the orientation during translational movements of the robot.

Another task where distributed sensors perform poorly is mapping, i.e. estimation of position of static landmarks in the space. Since both the sensors and landmarks are not moving the corresponding scans are always the same (except for the noise), so repeated measurements contain no additional information. Because of that, a map built using distributed sensors will in general be affected by calibration and other systematic measurement errors.

Onboard sensors do not have problems in map building owing to the mobility of the robot and therefore also the sensor. Direct estimation of the robot orientation based on detected landmarks is also possible. But, one notable weakness when compared to distributed sensors is the increased complexity of tracking moving objects like humans [14, 15]. Related to this issue are correlations between estimates, which are described in section 3.3.

3.2 System model and EIF estimation

The process model of the robot motion is given by $\mathbf{x}_k^r = f\left(\mathbf{x}_{k-1}^r, \mathbf{u}_k\right) + \mathbf{v}_k$, where the function f depends on the robot type (e.g. unicycle-like, differential drive, etc.) and $\mathbf{v}_k$ is the process noise, which is assumed to be zero-mean Gaussian with correlation matrix $\mathbf{Q}_k$. For implementing the estimator a Jacobian $\mathbf{F}_k$ of the function f is calculated at every step.

Landmarks are assumed static (i.e. $\mathbf{x}_k^l = \mathbf{x}_{k-1}^l$), whereas humans are assumed driven by Gaussian process noise: $\mathbf{x}_k^h = \mathbf{x}_{k-1}^h + \mathbf{v}_k$. In both cases the process Jacobian is equal to the identity matrix: $\mathbf{F}_k = \mathbf{I}$.

At each step the laser range finders give multiple measurements, one for each tracked object. These measurements are given with the distance and angle to object. The measurement equation is (here and in part of the equations below the time index k is omitted for simpler notation):

$$\mathbf{z} = h(\mathbf{x}) = \begin{bmatrix} q \\ tan^{-1}\left(\frac{\Delta x}{\Delta y}\right) + \theta_0 \end{bmatrix} + \mathbf{w}, \tag{1}$$

where:

$$\Delta x = x_0 - x_1, \tag{2}$$
$$\Delta y = y_0 - y_1, \tag{3}$$
$$q^2 = \Delta x^2 + \Delta y^2. \tag{4}$$

The subscript 0 stands for the sensor position, and 1 for the position of the detected object. For the distributed sensors the sensor position is fixed and known, wheres for the onboard sensor it is equal to the robot position and changes in time (we assume the onboard sensor is in the robot center, an extension to a different case is straightforward). $\mathbf{w}$ represents the measurement noise, which is assumed to be zero-mean Gaussian with correlation matrix $\mathbf{R}$.

Linearizing (1) the corresponding measurement Jacobian is obtained:

$$\mathbf{H} = \begin{bmatrix} \frac{-\Delta x}{q} & \frac{-\Delta y}{q} & 0 \\ \frac{\Delta y}{q^2} & \frac{-\Delta x}{q^2} & 0 \end{bmatrix}. \tag{5}$$

The last column corresponds to orientation, so it is excluded in human and landmark estimation. The zeros are reflecting the previously mentioned fact that the robot orientation cannot be directly detected with laser range finders.

For the onboard sensor the measurement depends on the position and orientation of the robot itself, so the measurement Jacobian has three additional columns that express this dependence:

$$\mathbf{H}_o = \left[\begin{array}{ccc|c} \frac{\Delta x}{q} & \frac{\Delta y}{q} & 0 & \multirow{2}{*}{$\mathbf{H}$} \\ \frac{-\Delta y}{q^2} & \frac{\Delta x}{q^2} & 1 & \end{array}\right]. \tag{6}$$

where $\mathbf{H}$ is the measurement matrix for the static sensor case, given by (5).

Using the above process and measurement models it is possible to implement the estimation algorithm. With the assumption of Gaussian noises and states a typical choice for the estimator is the extended Kalman filter (EKF). Here we use a EKF variant called the extended information filter (EIF) [22, 23].

EIF is based on the so called canonical parametrization of the Gaussian, which instead of the mean $\mathbf{x}$ and covariance $\mathbf{P}$ like in EKF, uses the information vector $\xi = \mathbf{P}^{-1}\mathbf{x}$ and information matrix $\Omega = \mathbf{P}^{-1}$. The prediction part of EIF is given by:

$$\mathbf{x}_{k-1} = \Omega_{k-1}^{-1}\xi_{k-1}, \tag{7}$$
$$\bar{\Omega}_k = \mathbf{F}_k\Omega_{k-1}\mathbf{F}_k^T + \mathbf{Q}_k, \tag{8}$$
$$\bar{\mathbf{x}}_k = f\left(\hat{\mathbf{x}}_{k-1}, \mathbf{u}_k\right), \tag{9}$$
$$\bar{\xi}_k = \bar{\Omega}_k\bar{\mathbf{x}}_k, \tag{10}$$

The bar denotes the predicted values.

In order to do the update new measurement information from all the distributed and onboard sensors needs to be combined. One way to do it is to send all the measurements to the fusion server. This however increases the computational burden of the fusion server. Instead, part of the calculations is done on the local nodes and the results are sent to to the server and fused.

For each object detected by the i-th sensor the vector $\mathbf{i}_i$ and its associated matrix $\mathbf{I}_i$ are calculated, which contain the new information from the measurement:

$$\mathbf{i}_i = \mathbf{H}_i^T\mathbf{R}_i^{-1}\left[\mathbf{z}_i - h\left(\bar{\mathbf{x}}\right) + \mathbf{H}_i^T\bar{\mathbf{x}}\right], \tag{11}$$
$$\mathbf{I}_i = \mathbf{H}_i^T\mathbf{R}_i^{-1}\mathbf{H}_i. \tag{12}$$

Note that feedback of the predicted state value $\bar{\mathbf{x}}$ from the fusion server is needed. This is due to the nonlinearity in the measurement equation (for the linear case it becomes: $\mathbf{i} = \mathbf{H}^T\mathbf{R}^{-1}\mathbf{z}$).

These values are sent to the server, where they are combined using a simple summation to obtain the final estimation values:

$$\Omega_k = \bar{\Omega}_k + \sum_{i=1}^{n}\mathbf{I}_{i,k}, \tag{13}$$

$$\xi_k = \bar{\xi}_k + \sum_{i=1}^{n}\mathbf{i}_{i,k}. \tag{14}$$

The advantage over EKF of a distributed fusion using EIF is obvious here. Although the prediction step (7)-(10) is more complex than that of EKF, the update step is much easier. It is reduced to a distributed calculation of (11) and (12) on the local nodes, which reduces the computation burden of the fusion server, which now needs to perform only summations. Additionally, the fusion server deals only with the information contained in $\mathbf{i}$ and $\mathbf{I}$, and does not need to know anything about the used sensors and measurement model. This information is used only by the local nodes, which makes distributed EIF suitable for easily combining different sensor types.

Some experimental results using EIF will be shown in comparison with Covariance Intersection in section 4.1.

3.3 Correlations between estimates

In the mobile robotics research it is well known that when the robot onboard sensors are used for robot localization, while at the same time landmark or human tracking is performed (as for example in SLAM - simultaneous localization and mapping), this leads to correlations between estimates. This affects all estimators based on the Kalman filter, and therefore also EIF. The fact that correlations between landmark estimates appear can be explained by the connection through the robot's pose: observing one landmark improves the estimate of the robot pose, and therefore also the previous poses, which in turn eliminates some of the uncertainty in the previously seen landmarks.

The problem that arises due to the correlation between estimates is that the number of parameters needed to describe the estimated variables increases. For uncorrelated estimates there is one state vector and covariance matrix for each tracked object. On the other hand, if estimates are correlated it is necessary to keep a vector that includes all the variables and a large covariance matrix that includes all the covariances plus the correlations between estimates. This is the reason why for uncorrelated estimates the number of parameters grows linearly with the number of tracked objects, whereas in the correlated case it grows exponentially [24].

The increased number of parameters is reflected in a higher computational load. Since it is necessary to send all the data through the network, the increase in the number of parameters needed to represent the estimate directly causes also an increase of the communication burden in the network. In addition, for each update not only the estimates for objects that were actually detected by the sensor, but estimates of all tracked objects need to be updated. Therefore the increase in the network load is actually much larger than it could be concluded just from the increase in the number of parameters. Also, the computational burden on the nodes is increased.

The rest of the paper analyses possible ways to deal with the correlation problem.

4 USING COVARIANCE INTERSECTION

Correlations between estimates can be explicitly avoided using the Covariance Intersection method. Covariance Intersection (CI) [25] has been proposed as a method of fusion of two estimates when the correlation between the estimates is unknown. It can also be shown that it is the optimal method of fusion in that case. CI also uses the Gaussian representation as the EIF, so the estimates are given by the state and the corresponding covariance matrix. Fusion of two estimates using CI is given by the following equations:

$$\mathbf{P}^{-1} = \omega \mathbf{P}_1^{-1} + (1-\omega)\,\mathbf{P}_2^{-1}, \tag{15}$$

$$\mathbf{P}^{-1}\mathbf{x} = \omega \mathbf{P}_1^{-1}\mathbf{x}_1 + (1-\omega)\,\mathbf{P}_2^{-1}\mathbf{x}_2, \tag{16}$$

where ω is a parameter between 0 and 1. Note that if we remove the ω and $(1-\omega)$ terms it results in the expression for convex combination, which corresponds to the Kalman and information filter algorithms.

It is possible to write the CI equations in a form compatible with the information filter update equations (11) and (12). The vector $\mathbf{i}$ and matrix $\mathbf{I}$ become:

$$\mathbf{i}_i = \omega\left(\mathbf{H}_i^T\mathbf{R}_i^{-1}\left[\mathbf{z}_i - h\left(\bar{\mathbf{x}}_i\right) + \mathbf{H}_i^T\bar{\mathbf{x}}_i\right] - \bar{\xi}_f\right), \tag{17}$$

$$\mathbf{I}_i = \omega\left(\mathbf{H}_i^T\mathbf{R}_i^{-1}\mathbf{H}_i - \bar{\mathbf{\Omega}}_f\right) \tag{18}$$

As with EIF, at each step from the new information is calculated based on new measurements, now using the equations (17) and (18), and sent to the fusion server. A nice characteristic is that there is no need to change anything on the fusion server, since the fusion algorithm remains completely the same. This enables easy inclusion of CI in the system of distributed sensors.

One good characteristic of CI is that since it treats all information as possibly being correlated, it prevents information reuse. Another good feature is that it is possible to obtain separate estimation for all tracked objects, thus giving uncorrelated estimates even when using onboard sensors. This is explained in [24], where CI was applied to the SLAM problem to achieve separate estimation of the robot and landmark positions.

Consequently, onboard sensors can be easily included in sensor networks, without introducing correlations between estimates. CI is used only for the onboard sensor, whereas distributed sensors continue to use the EIF method.

However, a weak point of CI is that, in contrast to the Kalman filter, it does not utilize all the available information. Hence the obtained estimate is somewhat conservative, as will be shown in the experimental results below.

4.1 Experimental results

Here we test the characteristics of EIF and CI estimators when applied to the sensing using both distributed and onboard sensors. The experiments were performed in a 5×7 meters experimental space that was covered with 4 distributed laser range finders Hokuyo URG-04LX. An ActivMedia Robotics Pioneer 2-DX robot equipped with a laser range finder of the same type was used and additionally 10 easily detectable landmarks were arranged in the space.

Fig. 5 shows the result of tracking the robot and mapping of landmark poses using EIF in a sample experimental

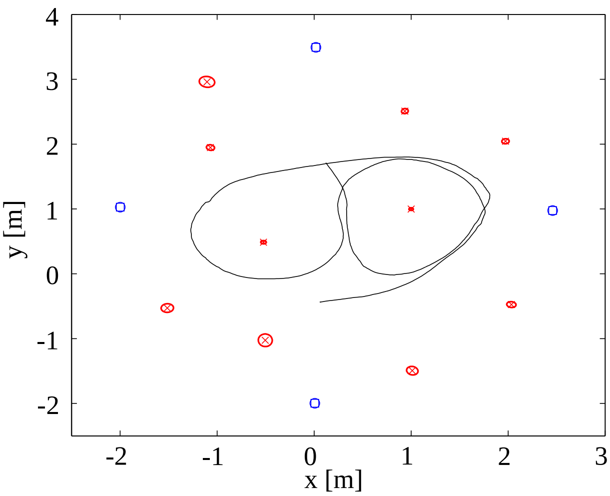

Fig. 5. Estimation result: the obtained robot path is shown, along with the estimated position of landmarks and their uncertainty ellipses; the position of static sensors is marked with squares

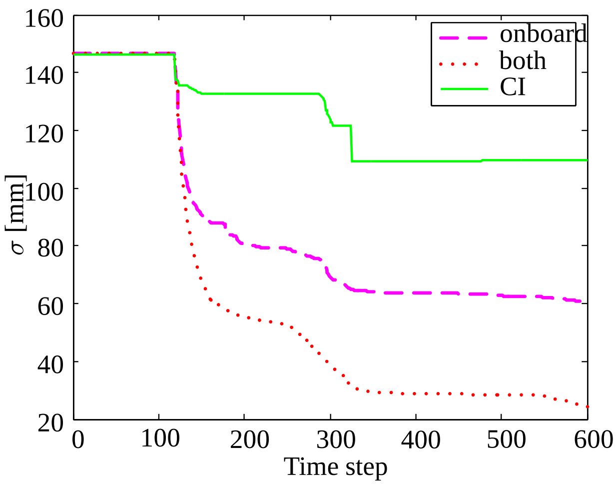

Fig. 6. Obtained landmark position uncertainty

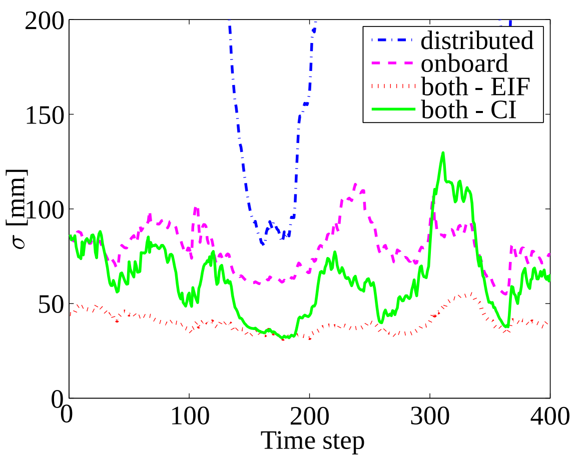

Fig. 7. Obtained robot position uncertainty

run. The ellipses show the uncertainty in the landmark estimates at the end of the experiment.

The results obtained with CI or only distributed sensors are similar. However, the difference between them can be easily observed in the resulting uncertainty (i.e. variance) of the estimates, which gives a measure of the estimate error. Fig. 6 shows the uncertainty in the position of one of the landmarks during the experiment while performing robot localization and mapping using different methods, namely EIF with only onboard sensors and both distributed and onboard sensors, and Covariance Intersection. As mentioned previously, distributed sensors are not appropriate for landmark tracking so they are excluded here.

When the onboard sensor is used the estimate started improving from the moment the landmark is first detected by the sensor. After that it steadily improved, even during the time the onboard sensor does not track the landmark, because of the correlation with other landmarks. Even though the landmarks were not tracked by the distributed sensors, the improvement over the estimate with only the onboard sensor is due to the fact that the estimate of the robot's position is better.

On the other hand, with CI the estimate improved only during the time it was actually detected by the robot's sensor. Since there were no correlation between different landmarks, updating one landmark did not give new information about any other. Also, although the mobile sensor detected the landmark after the 540th step, there was no improvement in the estimate due to the fact that the uncertainty of the measurement was larger than the uncertainty of the landmark.

Fig. 7 shows the uncertainty in the robot position estimate during the same experiment. As expected, tracking using only static sensors gave the largest uncertainty. On the other hand, when the information from the robot onboard sensor was also included using the EIF the best result of all proposed methods was obtained, reflecting the fact that all of the available information was utilized. The result obtained when only the onboard sensor was used lies somewhere in between.

It is obvious that Covariance Intersection gave a larger uncertainty than EIF as expected. Since for static sensors we continue to use the EIF, the obtained result is not as conservative as in the landmark case. Nevertheless, still at times it becomes even larger than the estimate using only the onboard sensor.

It can be concluded that Covariance Intersection provides an easy way to avoid the introduction of correlations between estimates. Since it is just enough to use CI instead of EIF (equations (17), (18) instead of (11), (12)) on each onboard sensor for processing the measurements it is very simple to use and the inclusion of onboard sensors in the network of static sensors becomes easy and straightforward.

However, estimation results show that CI gives rather conservative results. This leaves the question whether it is possible to deal with the problem of correlation in a different way, which is analyzed next.

5 SPECIAL CASES

Up to here a general estimation scheme that can be used for tasks of robot, human and landmark tracking and any combination of them was discussed. Here two important cases that are typical in the robot application in Intelligent Spaces are considered: the building of the map of the static environment, and tracking of moving objects (with an already known map).

5.1 Landmark mapping – hierarchical architecture

Here we consider the case when there is only one robot being tracked while at the same time the landmark positions are estimated. This is a situation that arises when a map of the landmarks needs to be built, as is often done prior to use of the robots inside a space.

The important thing to notice about this case is that only the measurements from the onboard sensor are used to update the landmarks positions. This is because, as previously mentioned, static sensors cannot be used to update the information on static objects.

For this reason, it is better to perform the updating the landmark position estimates directly on the robot, instead on the fusion server. In order to achieve this, a change in the fusion architecture is needed so a hierarchical architecture is introduced, Fig. 8. In this architecture the robot is placed above the fusion server in the hierarchy. The data from distributed sensors is first sent to the fusion server and the combined information is sent further to the robot. The robot implements the estimation and fusion algorithm and feeds back the estimated values to the fusion server, which in turn sends it to the distributed sensors. Again, feedback is needed to insure that all the linearizations at local nodes are done at the same point, so that the final result is the same as would be obtained with a centralized architecture.

In this architecture however, there is no need for the robot to send the fusion server and distributed sensors the estimates of the landmarks. Instead, they are dealt with locally on the robot side. The robot treats the information from the distributed sensors as additional measurements of its position, and implements the estimation of both its own pose and the position of landmarks.

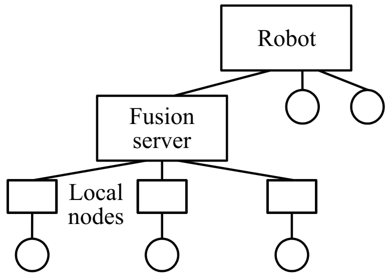

Fig. 8. Hierarchical distributed estimation

Nevertheless, in this case the correlations between estimates are not actually avoided. This can be seen as an external measurement aided SLAM, so some SLAM algorithm must still be implemented on the robot, e.g. the EKF-SLAM. Therefore, in this case reduction in the computation is not achieved, but what changed is the amount of data that needs to be sent through the network. Since only the robot estimates are communicated between the robot, fusion server and distributed sensors, this is significantly less data than if the whole estimate were sent.

5.2 Human tracking - sparsified correlations

The other situation where an alternative to EIF and CI is considered is the tracking of humans and robots (i.e. with fixed position of landmarks and calibrated distributed sensors). This is probably the most often used sensing application in iSpace.

An insight into the correlations in this case can be obtained from Fig. 9. It shows the uncertainty in the estimate of a human during an experiment in which it was intermittently detected by the onboard sensor.

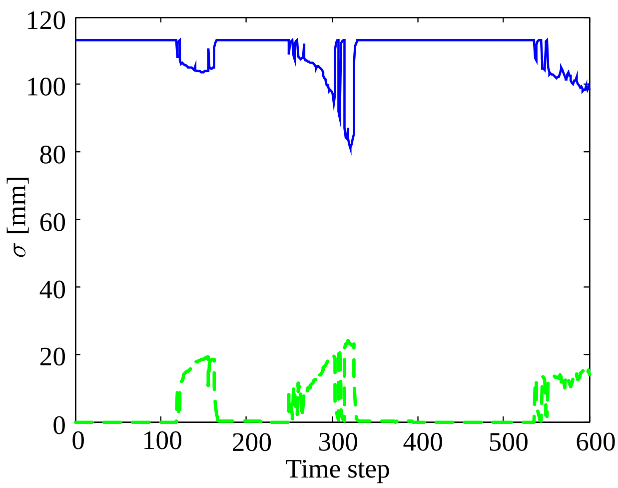

Fig. 9. Uncertainty in human tracking (solid line) and correlation between human and robot estimate (dashed line)

The obtained uncertainty shows that as soon as the onboard sensor stops tracking the human the estimate uncertainty goes back to the value obtained when using only distributed sensors. However, what is more important is that the value of the correlation between the robot and the human, which is also shown in the figure, goes toward zero. It never reaches exactly zero so actually a small correlation always remains. But this correlation becomes very small quickly after the onboard sensor stopped detecting the human.

This suggests the following trick: as soon as the value of the correlation between the object and the human goes below a certain threshold it is set to zero. If the threshold is not too big the change in the estimate results will not be very large.

In effect, what this does is turning the covariance matrix into a sparse matrix. Apart from the diagonal, the matrix has non-zero elements only for the elements connecting the robot and the humans that are currently detected (or have been detected until recently) by the onboard sensor. So, the estimation problem is divided into a correlated part – the estimates of the robot and the humans close to it, and uncorrelated part for other tracked humans, in which the estimation can be performed separately for each object.

Although this makes the estimates inconsistent, this inconsistency is small and diminishes with time. To illustrate this, the result of usual tracking was compared to the result when using the sparsification trick. The threshold was set to 10^{-5} mm, which in the considered case was reached after 10-20 steps (or 1-2 seconds) after the human disappeared from the onboard sensor scan. Fig. 10 shows the difference in the obtained uncertainty in the robot position.

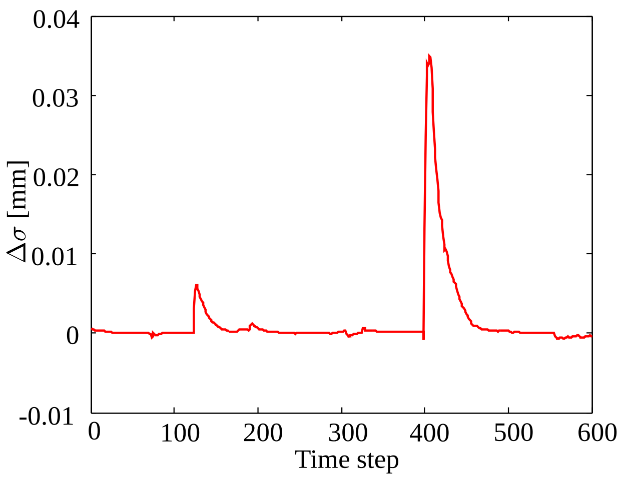

Fig. 10. Difference in obtained robot position uncertainty for standard and sparsified covariance matrix

Obviously the difference is very small. It is possible to notice the jumps in the difference at the points where sparsification was applied, however after that it gradually tends to zero. Note that even if this method were not applied in any implementation on a real computer the correlations would eventually become zero due to the numerical precision.

This method can also be applied if multiple robots are used. In that case for each robot one interconnected tracking area is obtained, so the whole estimation problem becomes divided into several correlated parts and possibly a part that is not covered by onboard sensors and is therefore uncorrelated. However care has to be taken so that correlations between robots do not occur, which can be obtained by assuring that any two robots do not track the same human or each other.

This method does not avoid correlations like CI, but only makes the problem less grave by dividing it into subproblems. But, as opposed to CI there is almost no change in the obtained result.

The improvement can be illustrated with a simple example. If we assume there are 2 robots and 9 humans, where each robot tracks 3 humans, while the remaining 3 humans are detected only by the distributed sensors, the number of needed parameters would be: 1034 in the fully correlated case (e.g. using EIF), 154 in the uncorrelated case (either with only distributed sensors or using CI) and 346 when using the sparsified correlation matrix. This is a significant decrease in the number of parameters.

6 CONCLUSIONS

The issue of sensing in Intelligent Spaces jointly using the static sensors distributed in the environment and the sensors onboard mobile robots was analyzed. It was shown how by combining the extended information filter and Covariance Intersection approaches it is possible to obtain a flexible distributed tracking scheme that allows fusion of all information while at the same time avoiding the increase in computation and communication due to correlations between estimates. In addition, special cases when it is possible to avoid the rather conservative results obtained when using CI were presented.

As summarized in Table 1 distributed sensors are useful in robot and human tracking, whereas onboard sensors can be used in robot tracking and map building, with the addition of human tracking when necessary. The question of how to structure the sensing is dependent on the specific problem. For example, one possible approach is also to use the onboard sensors only for robot localization, and discard all the measurement of humans. This way the correlation problem would be completely avoided. The robot sensors could also be used for tracking just one specific human, for example the user which requested a service, etc. In that case the increase in parameters due to correlations in the estimates of the robot's and that human's position would not be very significant.

REFERENCES

[1] L. Rudolph, **Project Oxygen: Pervasive, human-centric computing - an initial experience**, in Proceedings of the 13th International Conference on Advanced Information Systems Engineering (CAiSE '01), Interlaken, Switzerland, June 2001, pp. 1–12.

[2] J. Krumm, S. Harris, B. Meyers, B. Brumitt, M. Hale, and S. Shafer, **Multi-camera multi-person tracking for EasyLiving**, in Proceedings of the 3rd IEEE International Workshop on Visual Surveillance (VS), Dublin, Ireland, July 2000, pp. 3–10.

[3] B. Johanson, A. Fox, and T. Winograd, **The interactive workspaces project: Experiences with ubiquitous computing rooms**, IEEE Pervasive Computing Magazine, vol. 1, no. 2, pp. 67–74, 2002.

[4] C.D. Kidd, R. Orr, G.D. Abowd, C.G. Atkeson, I.A. Essa, B. MacIntyre, E.D. Mynatt, T. Starner, and W. Newstetter, **The aware home: A living laboratory for ubiquitous computing research**, in Proceedings of the Second International Workshop on Cooperative Buildings, Integrating Information, Organization, and Architecture (CoBuild '99), Pittsburgh, PA, USA, October 1999, pp. 191–198.

[5] J.-H. Lee and H. Hashimoto, **Intelligent Space – concept and contents**, Advanced Robotics, vol. 16, no. 3, pp. 265–280, 2002.

[6] S. Thompson, T. Horiuchi, and S. Kagami, **An environment driven model of human navigation intention for mobile robots**, in Proceedings of the 13th IASTED International Conference on Robotics and Applications, Wurzburg, Germany, August 2007.

[7] M. Shiomi, T. Kanda, H. Ishiguro, and N. Hagita, **Interactive humanoid robots for a science museum**, IEEE Intelligent Systems, vol. 22, no. 2, pp. 25–32, 2007.

[8] J.-H. Kim, Y.-D. Kim, and K.-H. Lee, **The third generation of robotics: Ubiquitous robot**, in Proceedings of the 2nd International Conference on Autonomous Robots and Agents, Palmerston North, New Zealand, December 2004.

[9] M. Broxvall, M. Gritti, A. Saffiotti, B.-S. Seo, and Y.-J. Cho, **PEIS Ecology - integrating robots into smart environments**, in Proceedings of the IEEE Conference on Robotics and Automation (ICRA '06), Orlando, FL, USA, May 2006, pp. 212–218.

[10] T. Sasaki and H. Hashimoto, **Hierarchical framework for implementation of Intelligent Space**, in Proceedings of the 33rd Annual Conference of the IEEE Industrial Electronics Society (IECON '07), Taipei, Taiwan, November 2007, pp. 28–33.

[11] D. Brščić and H. Hashimoto, **Model based robot localization using onboard and distributed laser range finders**, in Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '08), Nice, France, November 2008, pp. 1154–1159.

[12] B.K. Kim, N. Tomokuni, K. Ohara, T. Tanikawa, K. Ohba, and S. Hirai, **Ubiquitous localization and mapping for robots with ambient intelligence**, in Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '01), Beijing, China, October 2006, pp. 4809–4814.

[13] Y.-S. Kim, J.H. Lee, B.K. Kim, H. Do, T. Tomizawa, K. Ohba, X. Liang, K. Ohara, T. Sugawara, and T. Tanikawa, **Localization of ubiquitous environment based mobile robot**, in Proceedings of the International Conference on Control, Automation and Systems (ICCAS '07), Seoul, Korea, October 2007, pp. 2355–2358.

[14] D. Schulz, W. Burgard, D. Fox, and A. Cremers, **Tracking multiple moving targets with a mobile robot using particle filters and statistical data association**, in Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '01), Seoul, Korea, May 2001, pp. 1665–1670.

[15] C.-C. Wang, **Simultaneous Localization, Mapping and Moving Object Tracking**, PhD thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, USA, April 2004.

[16] A. Fod, M. Matarić, and G. Sukhatme, **A laser-based people tracker**, in Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '02), Washington DC, USA, May 2002, pp. 3024–3029.

[17] H. Zhao and R. Shibasaki, **A novel system for tracking pedestrians using multiple single-row laser-range scanners**, IEEE Transactions On Systems, Man and Cybernetics – Part A: Systems and Humans, vol. 35, no. 2, pp. 283–291, 2005.

[18] M. Bennewitz, W. Burgard, G. Cielniak, and S. Thrun, **Learning motion patterns of people for compliant robot motion**, International Journal of Robotics Research, vol. 24, no. 1, pp. 31–48, 2005.

[19] A. Panangadan, M. Matarić, and G. Sukhatme, **Detecting anomalous human interactions using laser rangefinders**, in Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '04), Sendai, Japan, October 2004, pp. 2136–2141.

[20] B. Jung and G. Sukhatme, **Cooperative tracking using mobile robots and environment-embedded, networked sensors**, in Proceedings of the IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA '01), Banff, AB, Canada, July 2001, pp. 206–211.

[21] A. Corominas Murtra, J.M. Mirats Tur, and A. Sanfeliu, **Efficient active global localization for mobile robots operating in large and cooperative environments**, in Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '08), Pasadena, CA, USA, May 2008, pp. 2758–2763.

[22] P.S. Maybeck, **Stochastic Models, Estimation, and Control**, Academic Press, 1979.

[23] S. Thrun, W. Burgard, and D. Fox, **Probabilistic Robotics**, The MIT Press, 2005.

[24] S.J. Julier and J.K. Uhlmann, **Using covariance intersection for SLAM**, Robotics and Autonomous Systems, vol. 55, no. 1, pp. 3–20, 2007.

[25] S.J. Julier and J.K. Uhlmann, **General decentralized data fusion with covariance intersection (CI)**, in D. Hall and J. Llinas, editors, Handbook of Data Fusion, ch. 12, pp. 12/1–12/25, CRC Press, 2001.

Dražen Bršćić received his BSc and MSc degree in electrical engineering from the University of Zagreb, Croatia, in 2000 and 2004, respectively, and the Dr.Eng. degree from The University of Tokyo, Japan in 2008. He is currently working as a postdoctoral researcher at the Technische Universität München, Germany. His research interests include mobile robotics, object tracking and intelligent spaces.

Hideki Hashimoto received the BSc, MSc and Dr.Eng. degrees in electrical engineering from The University of Tokyo, Japan in 1981, 1984 and 1987, respectively. He is currently an associate professor with the Institute of Industrial Science, The University of Tokyo. From 1989 to 1990 he was a visiting researcher at the Massachusetts Institute of Technology, Cambridge, USA. His research interests are control and robotics, especially advanced motion control and intelligent control. Dr. Hashimoto is a member of the SICE, IEEE, RSJ and IEEJ.

AUTHORS' ADDRESSES

Dražen Bršćić, Ph.D.
Institute of Automatic Control,
Technische Universität München,
Munich, Germany
email: drazen@tum.de

Prof. Hideki Hashimoto, Ph.D.
Institute of Industrial Science,
University of Tokyo,
Tokyo, Japan
email: hashimoto@iis.u-tokyo.ac.jp

Received: 2009-11-02
Accepted: 2009-11-27