

# We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900

Open access books available

186,000

International authors and editors

200M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index  
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?  
Contact [book.department@intechopen.com](mailto:book.department@intechopen.com)

Numbers displayed above are based on latest data collected.  
For more information visit [www.intechopen.com](http://www.intechopen.com)



# Robot Localization Using Distributed and Onboard Sensors

Dražen Brščić and Hideki Hashimoto  
*The University of Tokyo  
Japan*

## 1. Introduction

This chapter is dealing with the introduction of mobile robots inside environments with ubiquitously distributed sensors, specifically with the problem of robot tracking in such environments. There have been many researches recently on environments where both sensors distributed in the space and actuators such as mobile robots are used to support human users in various ways (Saffiotti et al, 2008; Hasegawa & Murakami, 2006). While there are several names that are often used for such spaces, here we refer to them as Intelligent Spaces (iSpaces) (Lee & Hashimoto, 2002). Intelligent Spaces have the purpose to provide human with both informative and physical services. Fig. 1 shows the concept of iSpaces. It relies on a network of devices to obtain information about the space and its state and to act inside it.

The primary function required by the Intelligent Space is the ability to track the position of objects in the space, most importantly that of humans and mobile robots. The issue of determining the robot's location has been dealt with a lot in the robotics community, mainly using onboard sensors. In Intelligent Spaces distributed sensors provide additional sources of information that can be used in the localization task. The issue of combining onboard and distributed sensors for robot localization has not achieved much attention yet.

A variety of sensors have been utilized for tracking in ubiquitous systems like iSpace (Hightower & Borriello, 2001). However, the tracking characteristics vary greatly from sensor to sensor, and this influences their choice. Some commonly used types of tracking systems are camera systems, tracking systems based on a combination of transmitters or receivers of ultrasound or electromagnetic waves, floor pressure sensors, distance sensors like laser range finders, etc. While all of these types of tracking systems have both advantages and limitations, here we use laser range finders as both distributed and onboard sensors for tracking in Intelligent Space.

Laser range finders are devices that determine the distance to an object or surface using laser rays. They give accurate measurements and recently some comparatively low priced sensors appeared. Due to the nature of applications of Intelligent Space, the tracking system is required to have a relatively high accuracy. It is also desirable that human users do not have to wear any special tags. Plus, the sensors should be easy to install and work with, and

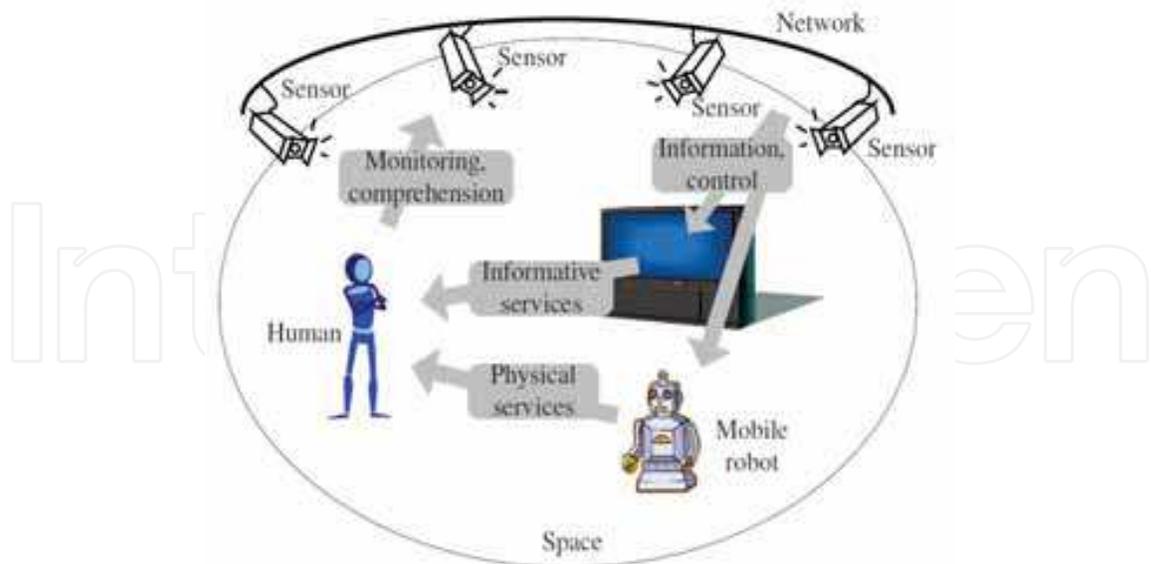


Fig. 1. Intelligent Space concept.

should be easy to apply for both human and robot tracking. Laser range finders satisfy all these requirements.

The research on position estimation using laser range finders is large. However, most of this work is on the self-localization of mobile robots equipped with onboard sensors. This also includes the extensive research on simultaneous localization and mapping (SLAM) (Thrun et al., 2005). Some recent works include also the detection and tracking of humans in the vicinity of robots, also using a laser scanner mounted on the robot (Schulz et al., 2001; Wang, 2004).

On the other hand, there are also several works dealing with objects tracking when the laser range finders are distributed in space, as is the case in Intelligent Spaces. In (Fod et al., 2002) static laser range finders were used for tracking people in everyday environments. In (Zhao & Shibasaki, 2005) the positions of pedestrians in large open spaces were estimated. Some researches used laser range finders to learn human motion patterns by extracting the trajectories of humans in the space (Bennewitz et al., 2005; Kanda et al., 2008).

This chapter tries to give the basic characteristics and methods for tracking using both distributed and onboard sensors. In the next section the basic characteristics of using distributed or onboard sensors and the advantages and disadvantages of using each of these sensing approaches are reviewed. Then, a centralized distributed architecture for Intelligent Spaces is proposed which provides flexibility and easy extension of its function.

Based on this architecture, methods for combined sensing using both robot's onboard sensors and sensors distributed in the space are described. The methods are tested in experiments where laser range finders were used as both onboard and distributed sensors.

First, tracking and fusion based on the Gaussian representations of the states (using an extended Information filter) is presented. Here the problem of correlations between the tracked objects which appears when onboard sensors are used is described and a solution based on the use of Covariance Intersection (CI) for onboard sensors is described, which explicitly avoids the correlation problem but gives conservative estimates.

Next, robot tracking based on the use of particle filters is presented. Here an extension of the popular Monte Carlo localization (MCL) algorithm that includes also the sensors distributed in the space is introduced. In addition, modeling of the robot geometry and mapping of the environment is addressed by extending the Rao-Blackwellized particle filter. In general these methods require centralized processing, so the issue of distributed sensing with so called Gaussian particle filters is also discussed.

## 2. Distributed and mobile sensors and fusion architecture

The application of sensors distributed in an environment to moving object tracking is straightforward and used in many applications. However, while they give very good results for tracking, distributed sensors are not very appropriate for building the map of the space. Since the sensors are not moving, the reading from static objects is always the same and further measurements give virtually no additional information. Attempts to build a map of the space with completely static sensors will in general be affected by calibration and other systematic errors. This is not the case with mobile robots equipped with onboard sensors since they make observations from different positions in space, so this types of errors average out.

Another problem with distributed sensors, especially for tracking with laser range finders, is that the orientation of the tracked object usually cannot be observed directly, since the 2D scan of the tracked object is usually small and lacking distinct features. In localization with onboard sensors this is in general not an issue since the used part of the scan is much larger. However, tracking of moving objects, such as humans, using only onboard sensors is usually more difficult than it is when distributed sensors are used. Even with a known map of the space, localization and moving object tracking can be challenging (Schulz et al., 2001). Also, distributed sensors offer continuous coverage of a tracked area, whereas continuous tracking of an object using onboard sensors has to be specifically implemented. A more psychological reason why distributed sensors are preferred is that they are embedded and not easily noticeable by users, whereas a following robot can make the users feel uncomfortable. These reasons justify why in most works on extraction of human walking patterns distributed sensors are used, e.g. (Bennewitz et al., 2005; Vasquez et al., 2008). The case of tracking humans was considered in our previous work (Brcsic & Hashimoto, 2007). Table 1 summarizes the most important points in sensing using only distributed or only onboard sensors. A combination of both distributed and onboard sensors, as considered here, can overcome the limitations of both types of sensors. Here we concentrate on the problems of robot tracking and mapping.

As explained in the previous section, the Intelligent Space is a complex system with multiple different sensors and actuators, and as such it is an important question how to design its architecture. In (Sasaki & Hashimoto, 2007) a hierarchical architecture of the iSpace as shown in Fig. 2 was proposed. It has a four layer structure consisting of sensors, information servers, applications and actuators. Information servers fuse the data from sensors and provide it to applications, which in turn send the commands to the actuators to realize various services. As discussed in (Sasaki & Hashimoto, 2007) this kind of hierarchical architecture makes it easy to add or change parts of the architecture without affecting the rest of the system, thereby providing the flexibility and modularity to iSpace. Because of that, we assume the use of this architecture throughout the chapter.

	Robot	Human	Map
Distributed	Good (orientation problem)	Good (and continuous)	Poor (non-informative)
Onboard	Good	Difficult	Good

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

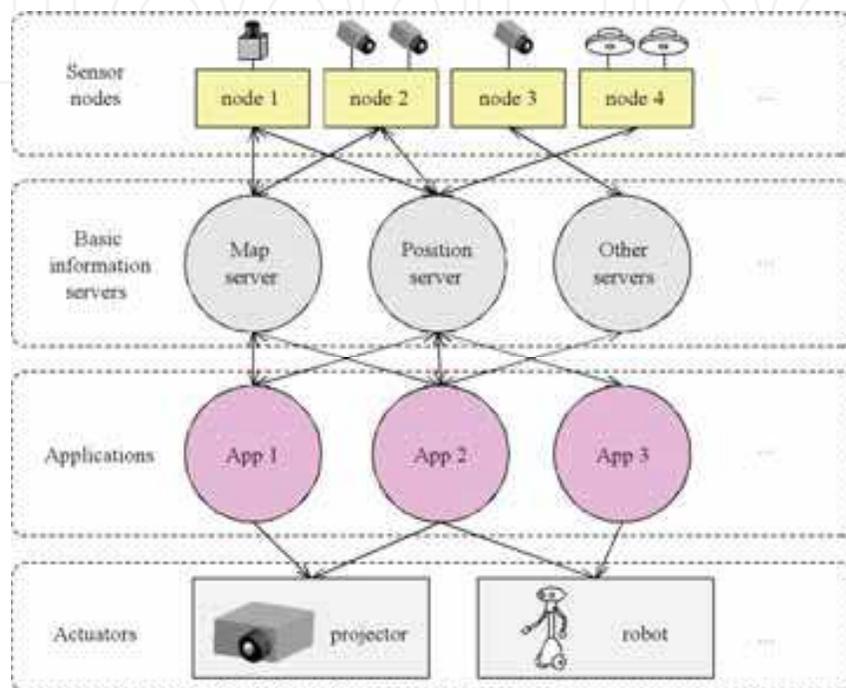


Fig. 2. Hierarchical architecture of Intelligent Space.

An important consequence of using this iSpace architecture is that for the fusion of sensor data a centralized fusion architecture is adopted, with an information server serving as the central processing unit. However, a pure central architecture where all the processing is done solely by the information server is not desirable, since computational and communication burden can be very large. Instead we make use of distributed processing, where the data is first processed on sensor nodes and then sent to the server.

### 3. Tracking with distributed and onboard laser range finders

We consider a system like the one depicted in Fig. 3. It consists of several distributed laser range finders distributed in the space and a mobile robot equipped with a laser range finder. Distributed sensors can observe the mobile robot, whereas both distributed and onboard sensors can observe other objects that serve as landmarks. It is assumed that the position of the landmarks is initially unknown.

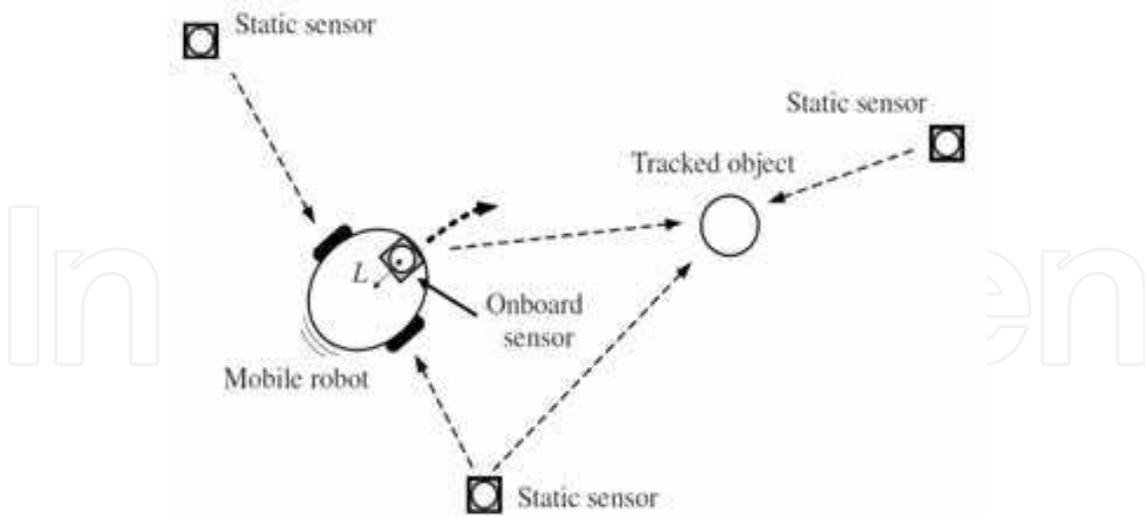


Fig. 3. Tracking system with both distributed and onboard sensors.

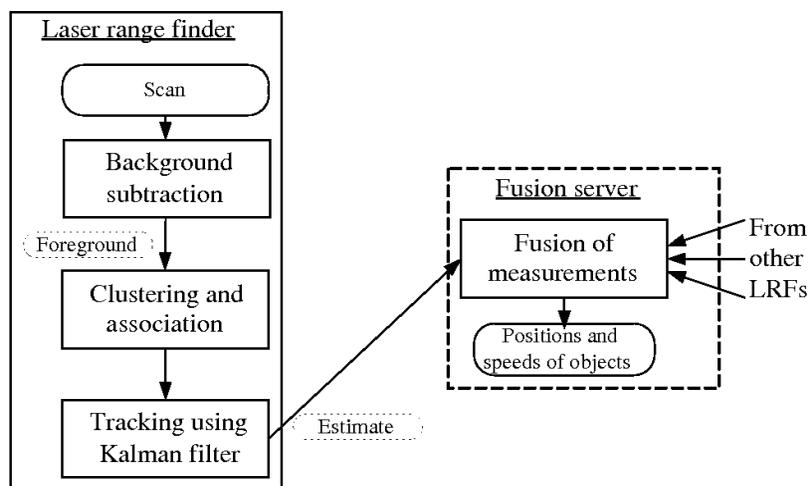


Fig. 4. Flow of object tracking using distributed sensors.

Figure 4 shows the flow of object tracking for the case of a distributed sensor. Each newly taken scan is processed through a number of steps and sent to the fusion server.

By comparing the new scan with the one where no moving object was present (the background scan), the parts that belong to moving objects can be extracted. These are further clustered and the clusters are associated to the tracked object. Finally an estimation algorithm is applied and the estimate is sent to the fusion server.

For onboard sensors the procedure is similar. From the scans landmarks are extracted from scans and used in the estimation of the robot's and landmarks' position, and this estimate is then sent to the fusion server. Any type of landmark can be used here, e.g. lines or corners – in our experiments we used artificial cylindrical landmarks which were easy to extract from the scans.

The laser scans give only the contours of the robot or landmarks, so for the use in the estimation algorithm the object centers are calculated first. This is done by calculating the center of the cluster as the mean value of the points in the cluster. Then the object center

position is approximated as being in the same direction from the sensor as the cluster center, but at a fixed larger distance than the, where the distance is dependent on the type of object (landmark or robot).

This approximation is simple, but can introduce errors if the shape of the robot or landmark are not circular, or in case of noise or misdetections. Also, for objects that are occluded by other objects, as often happens in cluttered or crowded environments, the approximation can give biased results. A different approach to using the object scans will be presented in section 4.

### 3.2 Estimation using the Information Filter (IF)

The described object tracking using a laser range finder gives as output the range and angle to the detected objects. The measurement model can be described as:

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

where  $\Delta x$ ,  $\Delta y$  and  $q$  are given by the following relations:

$$\Delta x = x_0 - x_1, \quad (2)$$

$$\Delta y = y_0 - y_1, \quad (3)$$

$$q^2 = \Delta x^2 + \Delta y^2 \quad (4)$$

The index 0 and 1 stand for the  $(x, y, \theta)$  coordinates of the sensor (either distributed or onboard) and observed object, respectively, whereas  $\mathbf{w}$  represents the measurement noise, which is assumed zero mean Gaussian with covariance matrix  $\mathbf{R}$ .

Linearizing the measurement equation gives the following measurement matrix for the distributed sensor case:

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

For the onboard laser range finder the position of the sensor is also changing with the robot's position, so the measurement matrix becomes:

$$\mathbf{H}_o = \begin{bmatrix} \frac{\Delta x}{q} & \frac{\Delta y}{q^2} & 0 \\ -\frac{\Delta y}{q} & \frac{\Delta x}{q^2} & 1 \end{bmatrix} \mathbf{H}, \quad (6)$$

where  $\mathbf{H}$  is the measurement matrix for the distributed sensors from eq. (5).

Using the measurement model given above together with the model of the robot and landmarks, which we omit here, it is possible to develop an estimator for the positions of the robot and landmarks. A standard approach is to use an extended Kalman filter (EKF). Here we use the dual form of the EKF – the Extended Information Filter (EIF) (Maybeck, 1979; Thrun et al., 2005). EIF gives the same result as EKF, but it has different computational characteristics which make it easy to be decentralized. Instead of representing the state with the mean  $\mathbf{x}$  and a covariance matrix  $\mathbf{P}$ , it uses the so-called canonical parametrization where the state is given by the information vector  $\mathbf{i} = \mathbf{P}^{-1}\mathbf{x}$  and information matrix  $\mathbf{I} = \mathbf{P}^{-1}$ . The full description of EIF can be found in the references and we leave it out here. In the distributed version each sensor separately calculates its information contributions:

$$\mathbf{i}_i = \mathbf{H}_i^T \mathbf{R}_i^{-1} \left[ \mathbf{z}_i - h(\bar{\mathbf{x}}_i) + \mathbf{H}_i^T \bar{\mathbf{x}}_i \right], \quad (7)$$

$$\mathbf{I}_i = \mathbf{H}_i^T \mathbf{R}_i^{-1} \mathbf{H}_i, \quad (8)$$

where  $i$  is the sensor index, the function  $h$  is defined by (1) and other values are as defined above. The variables with a bar represent predicted values.

These information contributions are sent to the fusion server, where they are combined into the final fused value using a simple summation:

$$\mathbf{i} = \bar{\mathbf{i}} + \sum_i \mathbf{i}_i, \quad (9)$$

$$\mathbf{I} = \bar{\mathbf{I}} + \sum_i \mathbf{I}_i. \quad (10)$$

### 3.3 Estimation using Covariance Intersection (CI)

However, as is well known from Kalman filter based solutions to the SLAM problem, the use of onboard sensor results in cross-correlations between the estimates. This means that the positions of different objects cannot be estimated separately, but instead all estimates must be lumped together in a large information vector and matrix. This in turn causes a significant increase in the computation, but also in the communication since the full estimates have to be transferred to and from the sensors.

To alleviate this problem we combine the use of the Information filter with the Covariance Intersection (CI) method. Covariance Intersection (Julier & Uhlmann, 2001) was 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. It also uses the Gaussian representation as the Kalman and Information filters, so the estimates are given by the mean state  $\mathbf{x}$  and the corresponding covariance matrix  $\mathbf{P}$ , or equally by the information vector and matrix  $\mathbf{i}$  and  $\mathbf{I}$ . Fusion of two estimates using CI is given by:  $\mathbf{I} = \omega \mathbf{I}_1 + (1-\omega) \mathbf{I}_2$  and  $\mathbf{i} = \omega \mathbf{i}_1 + (1-\omega) \mathbf{i}_2$ ,  $\omega$  being a parameter between 0 and 1.

It is also possible to write the CI equations in a form appropriate for use in the fusion equations (7) and (8). The vector and matrix representing new information become:

$$\mathbf{i}_i = \omega \left( \mathbf{H}_i^T \mathbf{R}_i^{-1} \left[ \mathbf{z}_i - h(\bar{\mathbf{x}}_i) + \mathbf{H}_i^T \bar{\mathbf{x}}_i \right] - \bar{\mathbf{i}} \right) \quad (11)$$

$$\mathbf{I}_i = \omega \left( \mathbf{H}_i^T \mathbf{R}_i^{-1} \mathbf{H}_i - \bar{\mathbf{I}} \right) \quad (12)$$

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 (Julier & Uhlmann, 2007), 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. Using (11) and (12) CI can be implemented for the onboard sensor, whereas distributed sensors use the EIF method (7) and (8).

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

Another problem of CI is that calculation of the  $\omega$  parameter usually involves optimization of some performance criteria, like the determinant of the fused matrix (Julier & Uhlmann, 2001), which makes the method more computationally involved when compared to EIF.

### 3.4 Experimental results

We made experiments in our experimental Intelligent Space. Part of the setup is shown in Fig. 5. Four Hokuyo URG-04LX laser range finders were used as distributed sensors and placed at the sides in a way that they cover the whole space (about 5x7 meters). A Pioneer 2-DX robot was equipped with the same laser range sensor in the front as onboard sensor. Ten cylindrical objects were used as landmarks.

Fig. 6 shows the result of estimating the robot and landmark positions after a sample run. The ellipses show the uncertainty in the landmark estimate. Also shown are the positions of distributed sensors.

While the position estimates are very similar for all estimation methods, differences can be seen in the obtained estimate uncertainties. Fig. 7 shows the uncertainty in the robot position estimate during the experiment. As expected, tracking using only distributed 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, and at times it even becomes larger than the estimate using only the onboard sensor. This is however achieved with a much smaller computation and communication load.

## 4. Geometrical model based method

In the method presented in section 3 all scan points belonging to an object were combined into one point to extract the center of the tracked object. This can be seen as an unnecessary loss of valuable information and a source of errors in the case when the object is occluded. Here we present another method that uses all these scan points and compares them with a geometrical model of the object. This approach results in a considerably different estimation method.

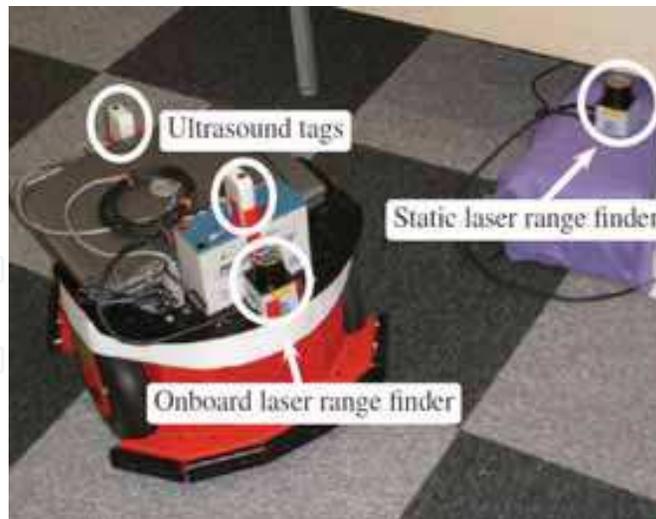


Fig. 5. Experimental setup: the mobile robot and one of the distributed laser range finders.

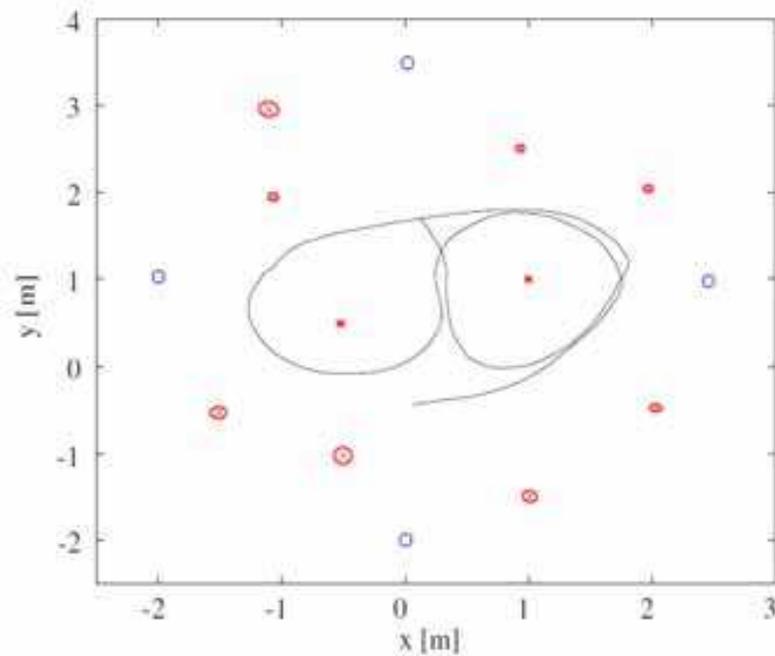


Fig. 6. Estimation result: the obtained robot path is shown, along with the estimated position of landmarks and their uncertainty ellipses; the position of distributed sensors is marked with squares.

**4.1 Modeling of the mobile robot and space**

Although several choices for the model of the robot exist, we use occupancy grids because of they can represent any kind of shape and are easy to build online. Occupancy grids have been very popular in the robotics community and can be easily combined with laser range finder measurements.

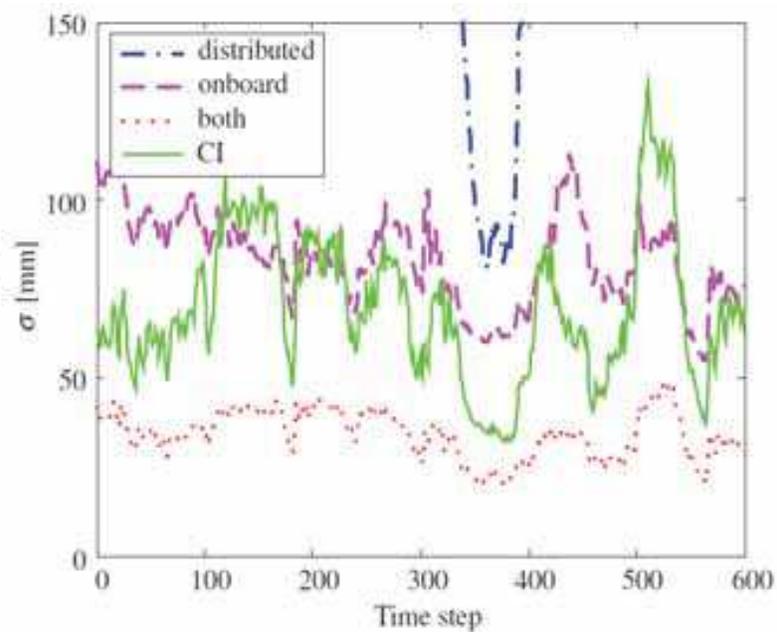


Fig. 7. Obtained robot position uncertainty when using the Information filter for only distributed, only onboard or both sensors, and when CI is used for onboard sensors.

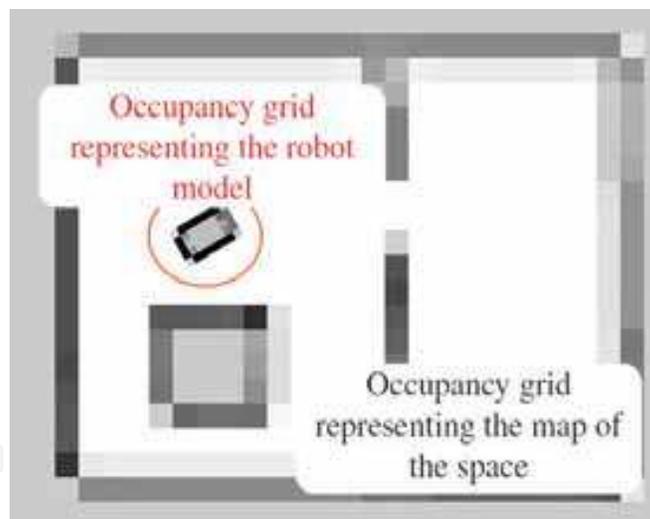


Fig. 8. Representation of the space and mobile robots with occupancy grids.

The idea of occupancy grids is to divide the space in a grid, usually evenly spaced. Each cell in the grid is given a measure representing the probability that the cell is occupied by an object. In order to make the combination easier, we model both the robot's shape and the geometry of the space using occupancy grids (i.e. instead of using landmarks like in the previous section here we use an occupancy grid representation of a map of all static objects in the space).

An example illustrating the overall model is given in Fig. 8. In addition to the occupancy grid representing the map of the space, there is also an occupancy grid that outlines the

shape of the robot. The space map is fixed to the world coordinate system, whereas the robot model changes its position with the robot.

#### 4.2 Estimation of the robot position

First we assume that the occupancy grids representing the robot and space models are known. A method how to build them will be explained in the next subsection.

Occupancy grids do not allow a straightforward usage of the Kalman filter or similar methods, like in the previous section. A common approach is to use a particle filter based approach (Arulampalam et al., 2002), and one standard particle filter based method used in robot localization with onboard sensors is Monte Carlo localization (MCL) (Dellaert et al., 1999).

The idea of MCL is to represent the belief on the robot's position by a set of particles, each representing one hypothesis on the current pose – i.e. the robots  $(x, y)$  position and orientation  $\theta$ . At every step the algorithm goes through several functions to update the set of particles, namely motion model sampling, particle weighting based on the measurement and resampling (Arulampalam et al., 2002).

In motion model sampling the position of each particle is updated based on the dynamic model of the tracked robot, given in (7). Next, for each updated particle a weight  $w$  is calculated, which describes a belief how well the new laser range finder measurement matches the occupancy grid map, given the updated pose (in our experiments we used the likelihood field method (Thrun et al., 2005)). Finally, the particles are resampled, i.e. a new set of particles is drawn from the current one based on the weights. This is repeated in each step with the new scans to obtain a new set of particles that defines the current belief on the robot pose.

Standard MCL uses only onboard laser range finders, however if the robot is also represented using an occupancy grid it is easy to extend it to distributed laser range finders. The only part that changes is the weighting of the particles. In addition to weighting based on the onboard sensor scan the particles are additionally weighted based on the matching of scans from distributed sensors with the robot model, and these new weights are used instead in the resampling.

#### 4.3 Building the robot model and space map

For the SLAM problem, a well known particle filter that builds up on MCL is FastSLAM (Montemerlo, 2003). This algorithm implements the same three functions in each update as described above for MCL, but in addition it has also a function that updates the occupancy grid. Similar to MCL, FastSLAM can also be easily extended to incorporate measurements from distributed sensors.

When building the model, not only the pose of the robot but also the log odds for each cell in the occupancy grid are unknown. A direct application of the particle filter would result in a huge number of necessary particles, but in FastSLAM the particles instead contain the robot pose and Gaussian representations of cell occupancy probabilities (Montemerlo, 2003). This is possible because of conditional independence of the map cells given the robot pose (this corresponds to the Rao-Blackwellization procedure from statistics, so this method is also commonly referred to as Rao-Blackwellized particle filter).

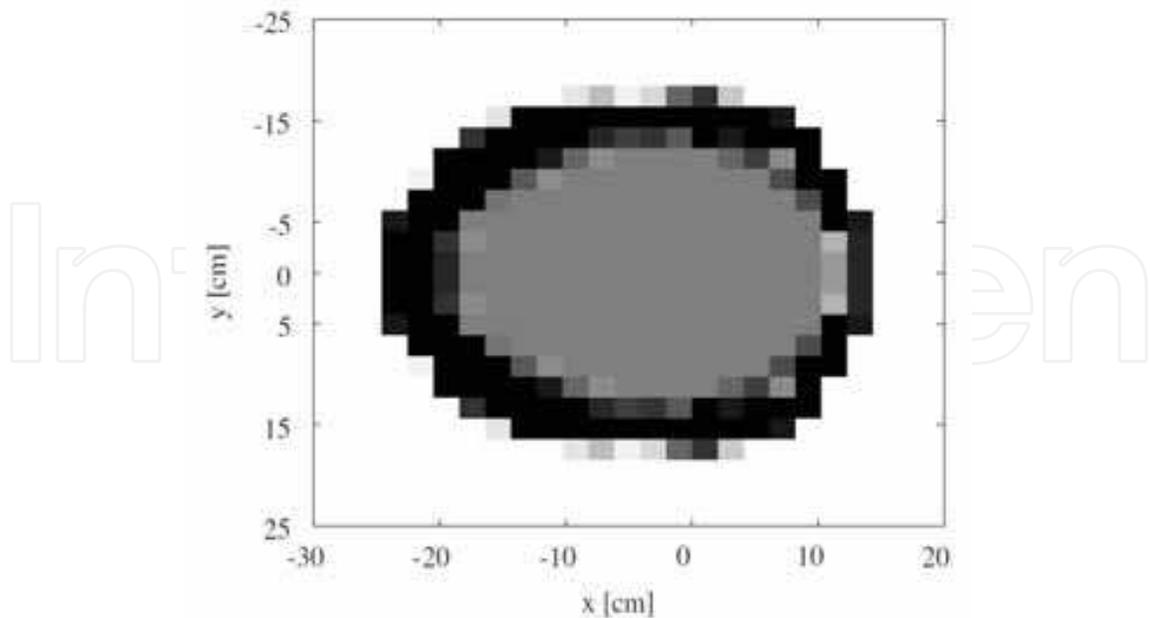


Fig. 9. Obtained occupancy grid representation of the robot model.

Since in our problem in addition to the map of the space we have also the model of the robot, both of these have to be contained in a particle. Due to the conditional independence of the robot models given the robot pose, a Gaussian representation of the cells can be used in the same way as for the map of the space. Therefore, each particle contains the robots pose, an occupancy grid representing the map of the space and an occupancy grid describing the robot model.

In addition to model sampling, particle weighting and resampling, at every step and for each particle both occupancy grids are updated. This is implemented using ray casting, where for each laser ray all the cells that it traverses are updated (Thrun et al., 2005).

We also use an assumption that the left and right side of the robot are symmetric, which is the case with most mobile robots. This simplifies the calculations since we only need half of the occupancy grid to represent the robot model.

#### 4.4 Experimental results

We made experiments in our Intelligent Space room using the same setup as in section 3.4. For the map of the space we used an occupancy grid with cells of size 10x10cm, whereas for the robot model we used 2x2cm cells.

Fig. 9 shows the occupancy grid representing the model of the robot obtained in the experiments using the method described above. This model matches closely the geometry of the real robot (the Pioneer 2-DX shown in Fig. 5), which has a width of about 30 cm, its front is at a distance of about 15 cm from the center, whereas the back is at about 25 cm.

Fig. 10 shows the obtained occupancy grid representing the map of the space.

Finally, we also made a tracking experiment, where the robot model and map of space obtained above were used. A comparison with the EIF method presented in section 3 is shown in Fig. 11. In order to get a measure of the precision of the proposed estimation method, the obtained estimation results are also compared with the measurements from the

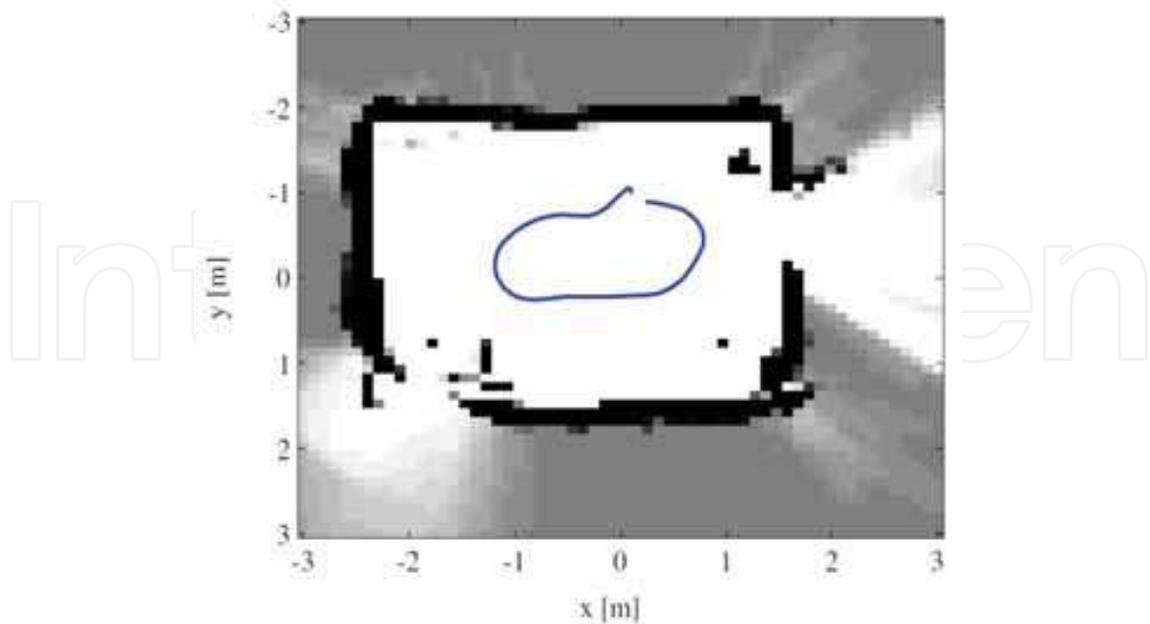


Fig. 10. Robot path and obtained map of the space.

ZPS ultrasound positioning system (by Furukawa Electric, Ltd.) that is installed in our experimental Intelligent Space.

We see that the two results are very close. By observing the differences between the estimates and the ultrasound system results, we noticed that throughout the experiment they mostly stayed below 5 cm for both methods. Overall the model based method gave slightly better results compared to EIF, which is expected since instead of an approximation it uses an accurate model of the robot.

#### 4.5 Distributed fusion using Gaussian particle filters

The model based tracking method discussed so far assumes that the data from all sensors are collected in a central place and processed simultaneously. However, as discussed also previously in section 2, centralized fusion is characterized by a large computational and communicational load, especially since all laser scans need to be transferred through the network.

Some approaches to distributed fusion of particle filters have been presented in e.g. (Rosencrantz et al., 2003; Ong et al., 2006), which use approximations of the particle sets, such as Gaussian mixture models. Here we investigate the approximation using simple Gaussian representations. One of the advantages of using particle filters is that they can deal with distinct hypotheses and non-Gaussian states easily, so representing particle sets with Gaussians might not seem justified. However, it can also be argued that due to measurements from distributed sensors the robot position can always be set to just one location, and therefore an unimodal distribution like the Gaussian can give a satisfactory representation.

Using the Gaussian representation of the state with a particle filter corresponds to the "Gaussian particle filters" (GPF) discussed in (Kotecha & Djurić, 2003). A distributed architecture is also assumed, where the sensors calculate new information from the

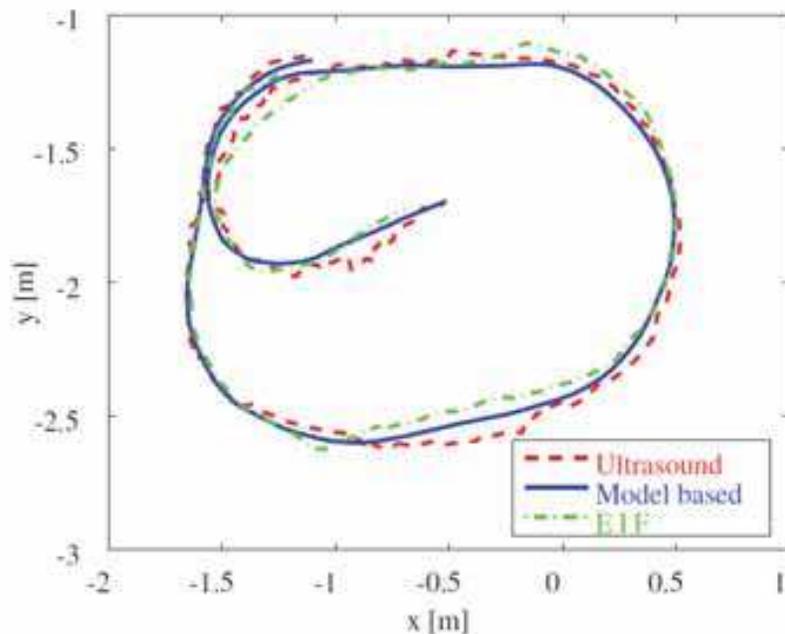


Fig. 11. Robot tracking results using EIF and the model based method shown in comparison to the ultrasound system tracking measurements.

measurements and send it to a fusion server, which in turn calculates the prediction step and feeds it back to sensors.

Since the Gaussian representation is used, the prediction on the fusion server can be performed in the usual way, based on the process model and the Kalman filter equations. The calculation on the sensor nodes is implemented in particle filter form, as follows:

- a set of particles is drawn from the predicted state vector and covariance matrix
- based on the measurements for each particle weights are calculated in the same way as in MCL or FastSLAM
- based on the particle states  $\mathbf{x}^{(n)}$  and weights  $w^{(n)}$  the updated state  $\mathbf{x}$  and covariance  $\mathbf{P}$  are calculated using the following equations:

$$\mathbf{x} = \frac{\sum_{n=1}^N w^{(n)} \mathbf{x}^{(n)}}{\sum_{n=1}^N w^{(n)}} \quad (13)$$

$$\mathbf{P} = \frac{\sum_{n=1}^N w^{(n)} (\mathbf{x}^{(n)} - \mathbf{x})(\mathbf{x}^{(n)} - \mathbf{x})^T}{\sum_{n=1}^N w^{(n)}} \quad (14)$$

At each step a new set of particles is drawn from the Gaussian representation of the predicted state. For each of these particles weights are calculated based on the measurement – for distributed sensors this is done based on the robot model, and for onboard sensor on the environment model. In the final step the particles are transformed back to the Gaussian

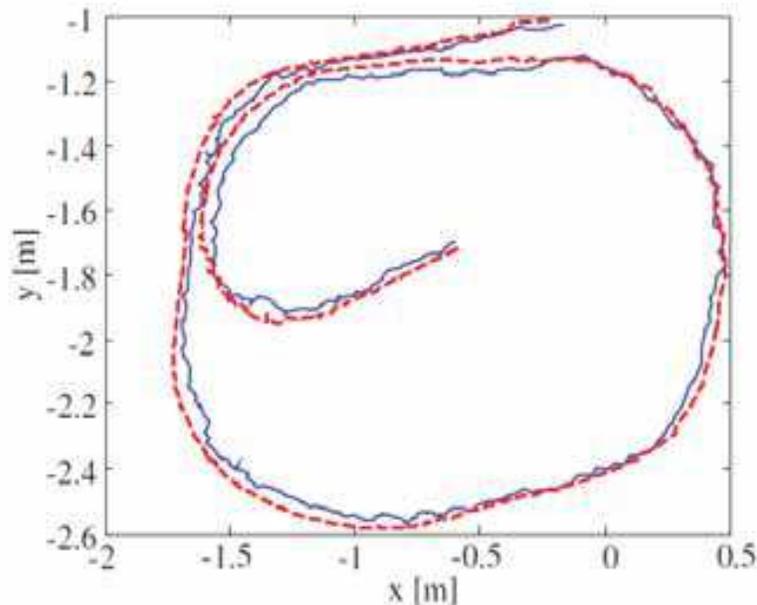


Fig. 12. Comparison of model based robot tracking results using distributed Gaussian particle filters (solid line) and standard particle filters (dashed line)

representation, by calculating their mean and covariance. This is done in the place of the resampling step of standard particle filters and is in general computationally less expensive. Fig. 12 shows the result of tracking a robot using the just described distributed Gaussian particle filter (same data as in Fig. 11 were used) compared with the result obtained using a centralized particle filter. Even though the GPF uses only an approximation of the particles for fusion, the obtained results are very close. The result is not so smooth due to the random nature of the algorithm and the fact that a small number of particles (50) was used.

Since in the distributed architecture all sensors work independently there is no need for all of them to implement the same tracking method. For example, onboard sensors could use EIF or CI as described in section 3, while distributed sensors use model based tracking.

## 5. Method comparison

The two methods presented here differ in their complexity. The first method determines the object center based on a simple calculation and then applies a Kalman filter based estimation method, whereas the second method uses a particle filter for the estimation. The particle filter methods in general have more computational demands, even though both methods can be made to run in real time on an average computer. A larger difference appears in the model building process, which for the second method is much more involved. For the first method model building corresponds to the inclusion of the estimation of the landmarks' positions, which as we said can be done uncorrelated with the robot position estimation and therefore very efficiently.

Yet, when the accuracies of the methods are compared there is not a large difference, as shown in the experiments. The reason for this is that even though the calculation in the first method is only approximate, the error introduced by the approximation is mitigated by

fusing the results from multiple sensors. Even though the second method uses a more exact algorithm, it does so at a cost of an involved computation.

Nevertheless, the advantage of using the model based method becomes more highlighted when used for tracking the robot in crowded or complex spaces. In these situation the occlusions in the sensor scans increase, thereby introducing an error in the approximation used in the first method. For onboard sensors it is obviously an advantage of using the whole scan instead of landmarks, which can be easily occluded.

Another question that arises in actual applications is the fusion of distributed sensors, since in iSpaces we deal with multiple sensors connected through a network. The first method can easily be implemented in a centralized architecture, with each sensor sending the extracted object center to the fusion node. But due to uncorrelated estimates, it can also be very simply turned into a distributed or decentralized algorithm, while keeping the network load relatively small. On the other hand, an exact solution for the second method admits only centralized processing, which in turn also requires significant communication loads since the whole parts of scan belonging to each object have to be sent. The Gaussian particle filter based estimation presented here gives an approximate way of obtaining a distributed estimation, but the characteristics and conditions under which it can be used still have to be studied.

## 6. Conclusion

This chapter discussed the issue of robot tracking inside Intelligent Spaces, where both sensors onboard the robot and sensors distributed in the space can be utilized. Two methods for robot tracking were presented, where a robot with an onboard laser range finder and several laser range finders at fixed locations in the space were used. One method uses an approximate relation for calculating the center of the tracked objects, and employs a combination of an extended Information filter and Covariance Intersection for position estimation. The other method extends the idea of occupancy grids to the modeling of the robot, and applies a particle filter based estimator for tracking and building of the robot model and map of the space. The methods are tested in experiments and a comparison is given.

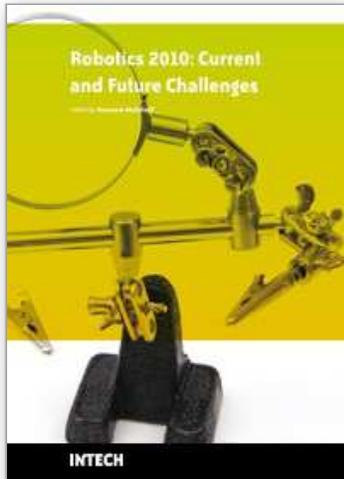
## 7. References

- Arulampalam, S.; Maskell, S.; Gordon N. & Clapp, T. (2002). A Tutorial on Particle Filters for On-line Non-linear/Non-Gaussian Bayesian Tracking. *IEEE Transactions on Signal Processing*, Vol. 50, 174-188
- Bennewitz, M.; Burgard, W.; Cielniak, G. & Thrun, S. (2005). Learning Motion Patterns of People for Compliant Robot Motion. *International Journal of Robotics Research*, Vol. 24, No. 1, 31-48
- Brscic, D.; Hashimoto, H. (2007). Tracking of Humans Inside Intelligent Space using Static and Mobile Sensors. *Proceedings of the 33rd Annual Conference of the IEEE Industrial Electronics Society*, pp. 10-15, Taipei Taiwan, Nov. 2007
- Dellaert, F.; Fox, D.; Burgard, W. & Thrun, S. (1999). Monte Carlo Localization for Mobile Robots. *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 1322-1328, Detroit, MI, USA, May 1999

- Fod, A.; Matarić, M. & Sukhatme, G. (2002). A Laser-Based People Tracker. *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 3024-3029, Washington DC, USA, May 2002
- Hasegawa, T. & Murakami, K. (2006). Robot Town Project: Supporting Robots in an Environment with Its Structured Information. *Proceedings of the 3rd International Conference on Ubiquitous Robots and Ambient Intelligence*, pp. 119-123, Seoul, Korea, Oct. 2006.
- Hightower, J. & Borriello, G. (2001). Location Systems for Ubiquitous Computing. *IEEE Computer*, Vol. 34, No. 8, 57-66
- Julier, S. & Uhlmann, J. (2001). General decentralized data fusion with covariance intersection (CI), In: *Handbook of Multisensor Data Fusion*, Hall, D. & Llinas, J. (Ed.) CRC Press
- Julier, S. & Uhlmann, J. (2007). Using covariance intersection for SLAM. *Robotics and Autonomous Systems*, Vol. 55, No. 1, 3-20
- Kanda, T.; Glas, D.F.; Shiomi, M.; Ishiguro, H. & Hagita, N. (2008). Who will be the customer?: a social robot that anticipates people's behavior from their trajectories. *Proceedings of the 10th international conference on Ubiquitous computing*, Seoul, Korea, Sep. 2008
- Kotecha, J. & Djurić, P. (2003). Gaussian particle filtering. *IEEE Transactions on Signal Processing*, Vol. 51, No. 10, 2592-2601
- Lee, J.H. & Hashimoto, H. (2002). Intelligent Space - Concept and Contents. *Advanced Robotics*, Vol. 16, No. 3, 265-280
- Maybeck, P. (1979). *Stochastic Models, Estimation, and Control*, Academic Press, ISBN: 978-0124807020
- Montemerlo, M. (2003). *FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem with Unknown Data Association*, PhD thesis, Robotics Institute, Carnegie Mellon University
- Ong, L.; Upcroft, B.; Bailey, T.; Ridley, M.; Sukkarieh, S. & Durrant-Whyte, H. (2006). A decentralised particle filtering algorithm for multi-target tracking across multiple flight vehicles. *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4539-4544, Beijing, China, Oct. 2006
- Rosencrantz, M.; Gordon, G. & Thrun, S. (2003). Decentralized sensor fusion with distributed particle filters. *Proceedings of the 19th Annual Conference on Uncertainty in Artificial Intelligence*, pp. 493-500, Acapulco, Mexico, Aug 2003
- Saffiotti, A.; Broxvall, M.; Gritti, M.; LeBlanc, K.; Lundh, R.; Rashid, J.; Seo, B.S. & Cho, Y.J. (2008). The PEIS-Ecology project: Vision and results. *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2329-2335, Nice, France, Sep. 2008
- Sasaki, T. & Hashimoto, H. (2007). Hierarchical Framework for Implementation of Intelligent Space, *Proceedings of the 33rd Annual Conference of the IEEE Industrial Electronics Society*, pp. 28-33, Taipei, Taiwan, Nov. 2007
- Schulz, D.; Burgard, W.; Fox, D. & Cremers, A. (2001). Tracking multiple moving targets with a mobile robot using particle filters and statistical data association. *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 1665-1670, Seoul, Korea, May 2001

- Thrun, S.; Burgard, W. & Fox, D. (2005). *Probabilistic Robotics*, The MIT Press, ISBN: 978-0-262-20162-9, Cambridge, MA, USA
- Vasquez, D.; Fraichard, T.; Aycard, O. & Laugier, C. (2008). Intentional motion on-line learning and prediction. *Machine Vision and Applications*, Vol 19, No. 5-6, 411-425
- Wang, C. (2004). *Simultaneous Localization, Mapping and Moving Object Tracking*, PhD thesis, Robotics Institute, Carnegie Mellon University
- Zhao, H. & Shibasaki, R. (2005). 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, 283-291

IntechOpen



## **Robotics 2010 Current and Future Challenges**

Edited by Housseem Abdellatif

ISBN 978-953-7619-78-7

Hard cover, 494 pages

**Publisher** InTech

**Published online** 01, February, 2010

**Published in print edition** February, 2010

Without a doubt, robotics has made an incredible progress over the last decades. The vision of developing, designing and creating technical systems that help humans to achieve hard and complex tasks, has intelligently led to an incredible variety of solutions. There are barely technical fields that could exhibit more interdisciplinary interconnections like robotics. This fact is generated by highly complex challenges imposed by robotic systems, especially the requirement on intelligent and autonomous operation. This book tries to give an insight into the evolutionary process that takes place in robotics. It provides articles covering a wide range of this exciting area. The progress of technical challenges and concepts may illuminate the relationship between developments that seem to be completely different at first sight. The robotics remains an exciting scientific and engineering field. The community looks optimistically ahead and also looks forward for the future challenges and new development.

### **How to reference**

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Drazen Brscic and Hideki Hashimoto (2010). Robot Localization Using Distributed and Onboard Sensors, Robotics 2010 Current and Future Challenges, Housseem Abdellatif (Ed.), ISBN: 978-953-7619-78-7, InTech, Available from: <http://www.intechopen.com/books/robotics-2010-current-and-future-challenges/robot-localization-using-distributed-and-onboard-sensors>

**INTECH**  
open science | open minds

### **InTech Europe**

University Campus STeP Ri  
Slavka Krautzeka 83/A  
51000 Rijeka, Croatia  
Phone: +385 (51) 770 447  
Fax: +385 (51) 686 166  
[www.intechopen.com](http://www.intechopen.com)

### **InTech China**

Unit 405, Office Block, Hotel Equatorial Shanghai  
No.65, Yan An Road (West), Shanghai, 200040, China  
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元  
Phone: +86-21-62489820  
Fax: +86-21-62489821

© 2010 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the [Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License](https://creativecommons.org/licenses/by-nc-sa/3.0/), which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.

IntechOpen

IntechOpen