LOCAL DEPENDENCY ANALYSIS IN PROBABILISTIC SCENE ESTIMATION

Thilo Grundmann, Robert Eidenberger and Raoul D. Zöllner

Siemens Corporate Technology
Information and Communications
81739, Munich, Germany
{thilo.grundmann.ext, robert.eidenberger.ext, raoul.zoellner}@siemens.com

ABSTRACT

A general solution to the problem of jointly estimating the state of multiple entities is regarded computationally challenging at the time. Most solutions are based on the application of wide assumptions of independence. In many situations and constellations of entities, this is sufficient and leads to high quality results. In some situations as occlusion for instance the assumption of independence is violated heavily resulting in considerable errors. The proposed approach considers local dependencies, allowing to increase the accuracy of the estimation punctually, depending on application requirements, such as high precision localization for grasping operations or rough precision for semantic localization.

1. INTRODUCTION

In many applications including mobile robotics, traffic surveillance, driver assistance system or automatic sports game analysis, the states of numerous objects have to be estimated to aggregate a model of the scene. When estimating the full joint state of multiple objects, physical constraints can be modeled into a perfect prior, which suppresses all constellations that are physically impossible. Unfortunately, the prior can not be determined with such high accuracy, so some of the physical constraints commonly remain unmodelled in such a probabilistic model. In addition to this, the curse of dimension makes it almost impossible to estimate higher dimensional state spaces jointly. One solution to cope with this, is to assume independence between all objects and estimate each object's state separately. This way the possibility to model those dependencies in the state is lost, and the system is biased by this model error. However, the impact of unmodelled dependency errors (UDE) differs widely, since the relevance of an error depends also on the required accuracy of a result given by the application.

This leads to the idea to improve the accuracy of the estimation by identifying the most distracting UDEs, with respect to the application, and to correct those on the basis of the background knowledge.

The remainder of this paper is organized as follows: A brief overview about the related work on Bayesian methods and their applications is given in Section 2. Section 3 describes example application scenarios, which illustrate the proposed approach. Section 4 addresses the proposed general approach of the Bayesian framework, which is able to cope with local dependencies. In Section 5 the punctual optimizations of the Bayesian update step and their illustration according to the example scenarios are presented. Finally, the paper closes with a short discussion about the estimation of dependencies in the proposed framework.

2. RELATED WORK

For estimation based on multiple measurements, a state of the art method is the Bayesian filter scheme [10]. There are different implementations, as the Kalman Filter [5] which assumes all distributions to be Gaussian and the motion and sensor model to be linear. Extensions to the Kalman Filter as the Unscented Kalman Filter [4] account for non-linearities in the models, whereas Sequential Monte Carlo methods generally have no restriction regarding the form of the distributions of the sensor and motion model.

Object localization that forms the measurement basis for a full scene estimation has been studied widely. Most of the approaches rely on certain properties of the objects as mono-coloring, texture or form. The key to universal scene model estimation seems to be a combination of multiple object recognition methods. The work of Azad et al.[1] describes an object localization method, that combines two different recognition methods, one is based on interest points as proposed in [8] and the other relies on untextured colored objects. This method is integrated in a robot, in order to localize objects with one measurement per grasping task.

The field of multiple object tracking is closely related to the task of estimating a full scene model consisting of different objects. Most of the implementations of the Bayesian filter scheme that have been used to track multiple objects, assume full independence between the objects and have subsequently the problem of data association. There are many approaches that solve the data association problem, which occurs in independent multiple target tracking as well as in scene modeling as PDAF, JPDAF[2] or Monte Carlo data association [6].

Rasmussen and Hager have proposed a multiple target tracking system, called Joint Likelihood Filter [9] that explicitly reasons about occlusion. Since their system estimates a 2D position the depth information of the targets is not included in the state, it is estimated separately. Kreucher et al. propose a multitarget tracking system [7], that estimates the joint multitarget probability density using a particle filter approach, leaving data association obsolete. They propose a measurement likelihood that considers the full state space and is able to handle occlusions. They do not reason about dependencies in the state, assuming that two targets can hold the same state. In contrast to that, this paper deals with the localization of real world objects that can not be located at the exact same spot.

3. APPLICATION SCENARIOS

Although, the proposed approach is generally independent of the application, we have chosen two example scenarios to better visualize the underlying principles and their effects. Both scenarios are part of an intelligent service robot application, which manipulates and transports objects and also interacts with humans.

The first example scenario shown in Figure 1 covers the area of scene analysis for manipulation of objects and for semantic space mapping tasks [11]. In this paper the main points to be considered are: 1. the scene is captured by sensors (e.g. a camera) and 2. it is composed of quasi-stationary objects (related to the update frame rate of the sensors). The issues we consider here, are the localization of objects (with variable accuracies) and the handling of occlusions and reflections as part of incomplete or disturbed sensor measurements.

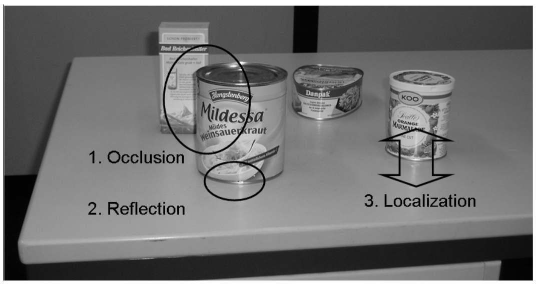


Figure 1: Scene perception and analysis. Local dependencies for the sensory detection and measurement of objects in terms of occlusion (1.), reflections (2.) and localisation (3.).

The second example scenario (Fig. 2) deals with the field of situation understanding by watching humans moving around [12]. The task to be considered here is the tracking of humans especially the prediction of their trajectory taking into account a 'willingness' to interact as shown in Figure 2.

Figure 2: Tracking persons (localization and movement). Using local dependencies for predicting the moving direction taking other persons and their ability to interact into account.

4. FRAMEWORK FOR PROBABILISTIC SCENE ESTIMATION

4.1. General Bayesian Filter Scheme

Considering the localization and tracking of objects using multiple measurements, the Bayesian filter is a state of the art method. To summarize shortly, one filter is applied for each object or entity and fuses together measurements (of different sensors) over the time.

Assuming a state x of an entity the filter update equations are:

$$\overline{bel}(x_t) = \int p(x_t|u_{t-1}, x_{t-1}) bel(x_{t-1}) dx_{t-1} \quad (1)$$

$$bel(x_t) = \eta p(z_t|x_t)\overline{bel}(x_t) \quad (2)$$

The first equation (1) is called prediction step and uses the believed state and in case of an entity that is able to actively change its state the control u to predict the intermediate believed state at the next time step. In the second equation (2) this intermediate belief is updated to the final new belief with the help of the measurement z. The term $bel(x_t)$ is an abbreviation for $p(x|z_t, z_{t-1}...z_0)$,while $p(x_t|u_t, x_{t-1})$ is commonly referred to as motion model and $p(z|x)$ is called measurement model.

When estimating the state of n entities $X^n = (x^1, x^2, ...x^n)$ and assuming independence between their individual states a single implementation of the Bayesian filtering scheme following the above equations is applied for each object.

When x^i describes the state space of a single entity and $x^{i,j}$ denotes the joint state space of two entities the following assumptions of independence can be stated:

$$bel({x^i}_t) = bel({x^i}_t|{x^j}_t) \quad (3)$$

$$p(z^i_t|{x^i}_t) = p(z^i_t|{x^{i,j}}_t) \quad (4)$$

$$p({x^i}_t|u^i_{t-1}, {x^i}_{t-1}) = p({x^i}_t|u^{i,j}_{t-1}, {x^{i,j}}_{t-1}) \quad (5)$$

As long as these independence assumptions are valid the separate Bayesian filter can be easily applied, in order to achieve good results (compare Section 2). However in most real world applications and also in our example scenarios these assumptions are violated quite often and whenever one of the above assumptions is violated, obviously the result of an independent estimation will be erroneous.

4.2. Local Dependencies in a Bayesian Framework

To cope with real world problems we suggest an extension of the general Bayesian filter scheme by considering dependencies, as stressed above, only temporarily and locally in the state space. In order to do so, the following conditions have to be fulfilled:

- The dependencies are temporarily limited. Otherwise, the dependencies should be permanently modeled[1].
- The dependencies can be modeled in the sense of one of the Bayesian update steps.
- The dependencies are detectable. Metrics exist, which can be used for their detection.
- For computational reasons and as a consequence of the detectability condition, the dependencies should be locally limited in the state.

For the cases, where these conditions are given the Bayesian update step can be extended (refer to Section 4.3), in order to consider the new dependencies as long as these are valid.

[1] In a SLAM situation for instance, the dependency of the map on the robots pose is neither temporarily nor locally limited. Hence, that dependency should be modeled permanently.

4.3. Local Bayesian Update Method for the Full Joint State

The overall idea of the method is to switch during the Bayesian update step from the independent to the full joint state as long as the dependency persists and afterwards to switch back to the independent state. It is possible to join arbitrary numbers of individual states if there are dependencies among them. For a better illustration let us consider two objects with states x^1 and x^2, which are mostly independent in their states. This state changes due to a manipulation task for a given time T_m before they become independent again. Further, let us assume that there is a function $F()$, which is able to detect the state dependencies (either by applying model knowledge or by passing the application's demand for higher accuracy) and a model describing the dependency based on a simulation function $sim()$. Then the algorithm performing the proposed method is composed of the following steps:

1. Perform Bayesian update assuming full independence of the states x^1 and x^2.
2. If state dependency is detected e.g. $F(x_1, x_2) == 1$ the states of x^1 and x^2 are joined according to (6).
$$bel(x^{i,j}) = bel(x^i)bel(x^j) \tag{6}$$
3. Extend the Bayesian update step by the additional dependency model according to (7). Since the distribution has to be normalized to sum up to one, the result of the $sim()$ function is normalized by $norm()$.
$$bel^s(x^{i,j}) = norm(sim(bel(x^{i,j}))) \tag{7}$$
In general the function $sim()$ incorporates physical laws or domain rules, which could be ignored most of the time. For instance, a simple implementation of $sim()$ would set the areas of the states of two rigid objects that imply an intersection to zero. As a consequence the belief states of the objects become more sharp.
4. The steps of the Bayesian update rule (8) (9) (the time indices have been dropped for clarity) can be executed multiple times.
$$\overline{bel}^s(x^{i,j}) = \int p(x^{i,j}|u, x^{i,j})bel^s(x^{i,j})dx^{i,j} \tag{8}$$
$$bel^s(x^{i,j}) = \eta p(z|x^{i,j})\overline{bel}^s(x^{i,j}) \tag{9}$$
Within the update rule, all motion and measurement models must be able to work correctly for the joint state space to fully utilize the extended state space. The joint state can be maintained as long as dependency is present, but the higher dimension (e.g. the state vector) accounts for highly increased computational costs.
5. When either the dependency is no longer existent or the accuracy requirement of the application is lowered, the joint space is marginalized back into the individual state spaces (10).
$$bel(x^i) = \int bel^s(x^{i,j})dx^j \tag{10}$$

As a result of this algorithm, it is possible to punctually increase the complexity and the accuracy of our estimation at the cost of temporarily higher computational demands. Obviously for any application, the benefit of extending the Bayesian update always has to be weighted against these additional costs in comparison to other methods that could increase the accuracy.

5. OPTIMIZATION BY A PARTIALLY BAYESIAN UPDATE

The maintenance of a full joint state over all parts of a Bayesian update is only reasonable, when strong dependencies occur in all parts of the update. Depending on a given scenario this is often not the case. Two wire frame objects for instance will almost never affect each other in the measurement likelihood, but will be strongly dependent in state space when close together. However, two persons on separated escalators will not be dependent in state space and they might occlude each other leading to a measurement likelihood that depends strongly on the state of the occluding person.

It is easy to construct an example for all possible combinations of dependencies. To account for this, it is appropriate to apply the minimum selection of the following methods, which is suited for the specific scenario. The idea is to analyze the dependencies according to the equations (3)(4)(5) and to partially extend the Bayesian update step only for the affected equation.

5.1. Virtual State Update

Instead of maintaining a joint state over a number of full Bayesian update circles as described before, it is also possible to perform only the first and the last part of the full joint state baysian update: The belief of two objects is transformed into a joint state space(6). Then with the help of a state dependency model $sim()$ the joint state is updated according to (7). This step may be based on a physical simulation, which checks the state against physical laws and the result of that update could be a clipping of the state regions that are impossible, or attenuating the ones that are physically unlikely.

Finally, the joint state is marginalized back into the individual state spaces (10). Whenever the dropped $bel(x^j)$ distribution is not equally distributed over the nonzero regions of $bel(x^i)$, the new found distribution will have been updated to a higher accuracy as long as the implications through the physical simulation have changed anything.

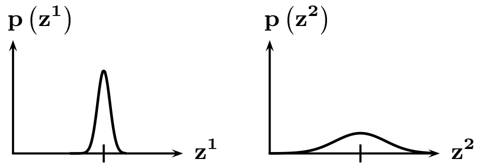
A detailed example is given, in order to clarify the proposed optimization by a virtual state update. The situation is related to the example scenario presented in Section 1, where the localization of objects, including a table and different types of cans and boxes, should be performed. A camera is used as a perception unit and the recognition is based on interest points to localize the small objects. Let us assume that the table has been localized more accurately beforehand and can be seen as persistent object in the scene. The described situation is shown in Figure 3.

In order to simplify the illustration we only examine the distribution along one dimension, in this case the z axis that describes the height. Both object's initial distributions in z are depicted in Figure 4.

For the virtual update the joint state is calculated according to (6). The resulting joint distribution (Fig. 5.1) is used as input for the simulation.

According to a physical simulation sectors in the state space, that are physically impossible, are detected. Here, these areas describe either an intersection between rigid objects or a floating can. In Figure 5.1, these areas are marked with two boxes. Note, that for illustrative reasons the physically plausible area is broadened instead of being infinitesimal small. The density value is set to zero, and the whole distribution is normalized to sum up to one (11).

Figure 3: Example scenario for the localization of two objects a table and a can, with different state (position) distributions along the z-axis.



(a) Table (b) Sauerkraut can

Figure 4: State estimate (z only) for the two objects can and table in the scenario illustrated in Figure 3

$$bel^s(x^{i,j}) = \frac{sim(bel(x^{i,j}))}{\int sim(bel(x^{i,j}))dx^{i,j}} \quad (11)$$

Finally, the joint distribution is marginalized into the marginal distributions according to equation 10. Figure 7 schematically depicts how the (z)-distributions of both objects are transformed into the same distribution, both having gained sharpness [2].

5.2. Joint Measurement Likelihood Update

Dependency in the measurement likelihood can occur in different ways. According to our application scenarios we will focus on two different types:

1. **The measurement likelihood of an object depends on the state and the measurement likelihood of another object.** That happens, when there is a coupling in the sensor for instance, leaving measurements dependent not only on the actual situation, but also on other measurements. Typically this is the case, during a scene perception with a camera where some objects are reflective. As a result in the image sensor some pixels having a high exposure bias the pixels nearby. The constellation of a very bright object and a second normal one that is located close to it couples the mea-

[2] In this situation, there is always a gain of sharpness for one distribution as long as the other one is not equally distributed over the non-zero parts of the first one. This holds, because the resulting distribution is the normalized multiplication of the two original distributions. To stop the filter from becoming overconfident conservative methods as Covariance Intersection [3] could be applied.

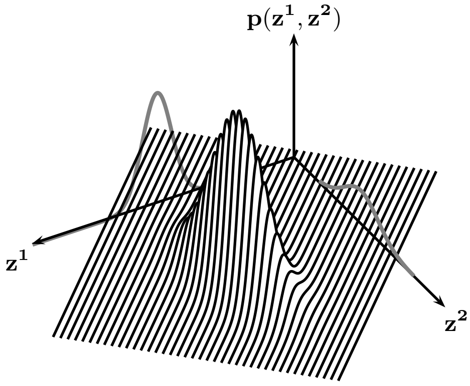


Figure 5: Joint belief distribution for table and can. The gray plots schematically depict the marginal distributions as in Figure 4.

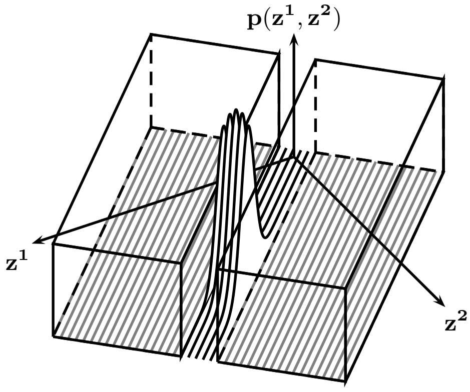


Figure 6: Adjusted joint state of the can and table through the virtual state update using a function $sim()$ based on a simulation of physical constraints. The boxes mark the physically impossible sectors.

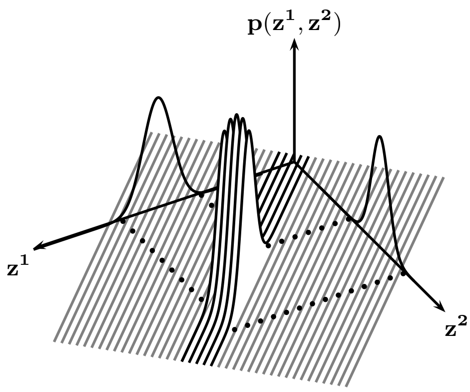


Figure 7: The marginalized distributions after the simulation, with a sharpened (z)-distribution of the can ($z_2 axis$).

surement likelihood of the beamless object on the measurement likelihood of the bright one. This can be considered by a measurement model that makes use of the joint state of the two objects (12).

$$bel(x^{i,j}) = \eta p(z|x^{i,j})\overline{bel}(x^{i,j}) \tag{12}$$

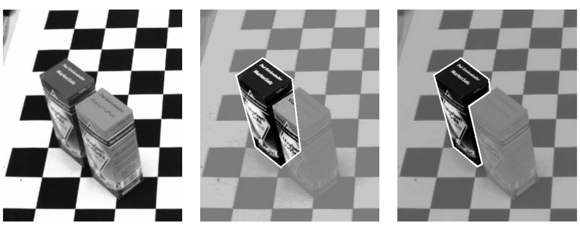
2. **The measurement likelihood of an object depends on the state of other objects.** This is always the case, when the presence of other objects changes the value or the propagation of the physical entity the sensor is based on. Common sensors that face this problem make use of electromagnetic waves (e.g. light, radar or laser) or sound which need a strictly defined and undisturbed propagation path to measure correctly (12).

$$bel(x^i) = \eta \int p(z|x^i, x^j)\overline{bel}(x^{i,j})dx^j \tag{13}$$

   The dependency in the measurement likelihood occurs independently of the dependency in the state or motion model. For example, using a camera, the measurement likelihood for one object, that is far from another and its state is therefore independent of the other object's state, can depend highly on the state of the other object in the case of occlusion. Of course, objects that are very close to each other tend to be dependent in state as well as in the measurement model. In the case of convex objects, the dependency due to occlusion occurs only for the occluded object.

To demonstrate the joint measurement likelihood update step, we chose a slightly different scenario. Two objects are located on a table, one occluding parts of the other one as seen from the camera. In this situation we focus on the object's distributions in y-axis, which is set to be the vertical direction of the check board pattern. We assume that the state of the unoccluded objects is known with high accuracy whereas the second object's distribution in y is quite uncertain.

For object recognition a simple interest point based method is used: Every interest point is seen as a binary sensor, that is able to either detect (=match) or miss its feature (=no match). The characteristics on true and false positives as well as true and false negatives are assumed to be equal for all binary interest point sensors and form their measurement likelihood. The location where the interest point is found in the sensor image is not taken into account here. For a given state, the measurement likelihood expects to detect certain features, based on the state. All of these binary sensor measurements are incorporated in the Bayesian update step.



(a) (b) (c)

Figure 8: Independent (b) and dependent (c) measurement likelihood for partly occluding objects (a).

In the situation of an independent measurement likelihood, the full number of detections will be expected for the occluded object. In the situation depicted in Figure 8(a), according to the measurement model it would be expected to detect features from the whole box, as depicted in Figure 8(b), but it will miss all that are occluded.

In the case of a joint likelihood measurement model (compare Figure 8(c)), we would expect less detections for the states that represent an occluded situation. The expectation for the states that correspond to situations without occlusion is the same for both models. Only in the case, that the situation contains an occlusion, both models differ in the resulting update. Updating the state using a joint likelihood measurement model will lead to a result where more probability mass is distributed over the occluded states, since the measurement likelihood assigns a higher probability to those states. This happens, because every detection that has not been measured but was expected for a state, lowers the probability of that state.

5.3. Joint Motion Model Update

Essentially, the principles stressed above can be applied also for the motion model, were dependencies on other object states can occur independently and consequently for the Bayesian update step, only the prediction step must be modified.

Our second example scenario, which tackles the tracking of humans in a situation understanding context the motion model of each entity could influence the prediction of another one's motion. As for the virtual state update a detection function $F()$ can be used, in order to trigger the motion dependency treatment. Looking very abstract at the human tracking scenario the tracked entities do not have any knowledge of the control of other entities neither do they have a precise motion model for other entities. It can be stated, that it is possible to condition the joint motion model only on the states of other entities (14), but also on the control (15) and the motion model (16) of other entities.

$$\overline{bel}^s(x^i) = \int p(x^i|u^i, x^i, x^j)bel^s(x^{i,j})dx^{i,j} \tag{14}$$

$$\overline{bel}^s(x^i) = \int p(x^i|u^i, u^j, x^i, x^j)bel^s(x^{i,j})dx^{i,j} \tag{15}$$

$$\overline{bel}^s(x^{i,j}) = \int p(x^i|u^{i,j}, x^{i,j})bel^s(x^{i,j})dx^{i,j} \tag{16}$$

In the person tracking scenario a realistic triggering function $F()$ depends on the state (e.g. relative position) of the persons, their motion directions and velocities as well as their internal states characterized by an attribute like 'willingness to interact' and the related interaction class. When two entities are walking towards each other and their internal state is 'willing to interact' the joint motion update according to the equations (14), (15) and (16) is triggered. As a result the motion of the tracked persons does not follow the physical (uniform) motion equations any more. According to the interaction type the motion slows down or stops at all as shown in Figure 2 (3.) in the case of a salutation with handshake.

6. ESTIMATION OF UNMODELED DEPENDENCIES

The proposed Bayesian update including all optimization facets is based on the assumption that a trigger function $F()$ exists, which detects whether the assumption of independence is violated or not.

It is obvious that it is impossible to find a general definition for $F()$, since it is highly related to the model used in the application. However, when designing a certain application based on the Bayesian framework the definition of such trigger functions as well as the corresponding models for the extended Bayesian update can be treated as special cases of the application. To recall, in our application scenarios we had presented some very basic trigger functions like:

- A simple estimator for state dependency evaluates the distance between two objects and considers the variance of that distance.
- For the measurement likelihood dependency in the case of a 2D camera, the estimator uses the distance from the object that is located closer to the camera to another object's line of sight.
- In the prediction dependency the estimator could also use the absolute distance between two entities in relation to their velocity and internal state to estimate their possible dependency in the prediction of their states.

In the literature more sophisticated metrics of dependencies can be found, which are strongly correlated to the application. For instance, in the field of learning manipulation tasks the relation between objects depends not only on their position and class but also on their internal state, which considers the role of the objects during a manipulation task [13].

Finally, we would like to stress the issue of complexity, when considering dependencies during the Bayesian update step. Of course the best estimator is the full calculation of the joint space, measurement likelihood and motion model, in practice however the required computational resources often exceed the capacity of a system. Therefore, ignoring or optimizing the level of dependency is a very important feature of such systems.

7. CONCLUSION

In this paper a general approach of a Bayesian framework for estimating a scene model is presented, which is able to cope with locally limited dependencies in the state space, measurement likelihood and motion model. We focused especially on the Bayesian update step, and showed how the local dependencies can be considered and what the resulting implications are. Further, the concept of punctual optimization by a partial Bayesian update was motivated, discussed and illustrated along the two defined application scenarios: 'scene analyzing' and 'person tracking'.

Due to the fact, that the main idea behind this approach is to join and disjoin state spaces only when the constellation of targets implicates that the assumption of independence is violated to a certain extent, and that this violation affects the state significantly in relation to the desired accuracy, this concept is able to improve its quality on demand, whilst coping with the curse of dimensionality.

8. ACKNOWLEDGMENT

This work was partly funded as part of the research project DESIRE by the German Federal Ministry of Education and Research (BMBF) under grant no. 01IME01D.

9. REFERENCES

[1] AZAD, P. ; ASFOUR, T. ; DILLMANN, R. : Stereo-based 6D Object Localization for Grasping with Humanoid Robot Systems. In: *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*. San Diego, CA, USA, 2007

[2] BAR-SHALOM, Y. ; XIAO-RONG, L. : *Estimation and Tracking: Principles, Techniques and Software*. London : Artech House Publishers, 1993. – 432–436 S

[3] JULIER, S. J. ; UHLMANN, J. K.: Using covariance intersection for SLAM. In: *Robot. Auton. Syst.* 55 (2007), Nr. 1, S. 3–20. – ISSN 0921–8890

[4] JULIER, S. ; UHLMANN, J. ; DURRANT-WHYTE, H. : A new approach for filtering nonlinear systems. In: *Proceedings of the 1995 American Control Conference, Seattle* (1995), S. 1628–1632

[5] KALMAN, R. : A new approach to linear filtering and prediction problems. In: *Transactions of the ASME, Journal of basic Engineering* (1960), Nr. 82, S. 34–45

[6] KARLSSON, R. ; GUSTAFSSON, F. : Monte Carlo data association for multiple target tracking. (2001)

[7] KREUCHER, C. ; KASTELLA, K. ; HERO, A. : Multitarget tracking using the joint multitarget probability density. In: *IEEE Transactions on Aerospace and Electronic Systems* 41 (2005), S. 1396– 1414

[8] LOWE, D. G.: Object recognition from local scale-invariant features. In: *International Conference on Computer Vision*. Corfu, Greece, September 1999, S. 1150–1157

[9] RASMUSSEN, C. ; HAGER, G. D.: Probabilistic Data Association Methods for Tracking Complex Visual Objects. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 23 (2001), Nr. 6, S. 560–576

[10] THRUN, S. ; BURGARD, W. ; FOX, D. : *Probabilistic Robotics*. MIT Press, Cambridge, MA, 2005

[11] VASUDEVAN, S. ; SIEGWART, R. : A bayesian conceptualization of space for mobile robots. In: *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on* (Oct. 29 2007-Nov. 2 2007), S. 715–720

[12] ZAJDEL, W. ; ZIVKOVIC, Z. ; KROSE, B. : Keeping Track of Humans: Have I Seen This Person Before? In: *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on* (18-22 April 2005), S. 2081–2086

[13] ZOELLNER, R. ; DILLMANN, R. : Using multiple probabilistic hypothesis for programming one and two hand manipulation by demonstration. In: *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2003