SLAM in the Dynamic Context of Robot Soccer Games

Stefan Tasse, Matthias Hofmann, and Oliver Urbann

Robotics Research Institute
Section Information Technology
TU Dortmund University
44221 Dortmund, Germany

Abstract. This paper evaluates the benefits of modeling the dynamic environment of robot soccer games as a SLAM problem. Moving objects such as other robots and the ball are not only tracked individually, but modeled in a full state and used for localization at the same time. This is described as an implementation of an efficient system capable of running in real time on limited platforms such as the humanoid robot Nao, and the system's benefit is evaluated using real world experiments.

1 Introduction

For an autonomous robot, knowing its location and the current state of the environment is essential as it represents the basis for planning and reactive behavior. Generally, this task may involve different aspects: Localization in known environments, mapping previously unknown environments, and tracking dynamic elements therein. In most common literature [1], those aspects are handled separately. Either a robot is only expected to localize in a known structured environment, leaving a choice among many existing solutions which mostly differ in their suitability for different kinds of uncertainties occurring during the robot's operation, or no prior knowledge exists at all as assumed in simultaneous localization and mapping (SLAM) scenarios, which represents the other extreme. Dynamic elements are generally ignored in both cases, either handled as noise or explicitly filtered out [2, 3].

Real world applications, however, always consist of a mixture of all those aspects. Many features in the robot's area of operations will be known beforehand, either from floor plans for indoor, or aerial photographs for outdoor scenarios, or from previous mapping efforts or other given specifications. On the other hand, such a priori information rarely covers all features which are of interest for the localization task. Some might even be occluded due to recent changes in the environment. Incorporating new information into a robot's internal map can therefore often improve its ability to localize precisely. Similarly, it can be shown that explicitly incorporating dynamic features into the system improves the estimation quality, both for the tracking result and the localization [4].

In the course of this paper the modeling of the full state in dynamic situations will be covered. Section 2 gives an analysis and a brief overview over related work.

Section 3 describes the actual implementation of such a system with the practical approximations necessary to reduce the problem's complexity to a point where it is applicable in real-time on limited embedded platforms, specifically on the Nao V3.3 robot. An evaluation is given in section 4 in the robot soccer context of the RoboCup Standard Platform League (SPL), and the paper is concluded in Section 5.

2 Modeling

This section addresses the different aspects of the problem in the context of a team of cooperating autonomous agents acting in highly dynamic competitive environments, therefore settling certain design choices as the basis for the implementation of the system described in section 3.

2.1 Localization and Robot-centric Tracking

The computational complexity of common filtering approaches naturally increases with the state's dimensionality. Separating the estimation of the robot's own position and that of the positions of other surrounding elements is therefore motivated from a performance standpoint. Moreover, tracking dynamic objects by stationary observers is a widely explored problem. Those are the main reasons why the tracking problem for autonomous robots is commonly done in a robot-centric local coordinate system.

The modeling of the localization and tracking aspects as a unified global estimation problem, however, has some advantages compared to simple tracking in robot-centric local coordinates. The latter necessitates the update of all tracked objects with odometry data, which e.g. for humanoid robots can be extremely unreliable. Those propagate nearly uncorrected into infrequently observed targets, leading to significant drift. At the same time, the separate robot pose estimation already corrected part of the odometry's errors and prevented the same drift in its estimate. This motivates the advantage of modeling dynamic objects in global reference systems. Additionally, information about tracked objects can be beneficial for the localization when modeled in a unified state. Even if the objects' motion uncertainty prevents their use for accurate localization, shared information about those objects among a team of agents might still resolve multi-modal or symmetrically ambiguous localization states.

This leads to a heterogeneous system which has some resemblance to the SLAM problem, since new features are mapped and used for localization at the same time. In previous literature pure localization and the full SLAM problem have been mostly separated. Only recent publications began exploring the intermediate between those extremes by incorporating a-priori information into systems otherwise formulated as SLAM, e.g. in [5] where a SLAM approach is augmented by a-priori information in form of aerial images. Moving objects are just considered to be obstructive in normal SLAM algorithms and either handled as noise [2] or tracked in separate model to be filtered out of the SLAM input, therefore without any direct positive effect on the SLAM output [3].

An alternative system providing the same characteristics as the one proposed here has already been published in [4]. It describes an adaptation of the Fast-SLAM concept to include a-priori information as well as newly mapped static and dynamic features with different degrees of uncertainty in their recognition processes and motion models. The approach presented here differs from the one in [4] in its vast use of Kalman filters in all different stages of the system, whereas the latter integrates the SLAM aspects by use of a Rao-Blackwellized particle filter. Notably, the system presented here can run in real-time on a Nao robot in parallel to all other modules necessary to participate in RoboCup SPL games.

2.2 Heterogeneous Information Sources

As stated so far, the proposed system should use information about previously known and previously unknown, static and dynamic features, and incorporate all those into a coherent estimate of both the robot's own positions as well as the potentially dynamic states of the other objects. This obviously implies various different, and more importantly, heterogeneous information sources.

Distinctions can be made according to the characteristics of each feature, whether it can be used for localization directly or needs to be mapped, too, either as a static but previously unknown feature or a dynamic one including motion updates. A features associated uncertainty can vary both with respect to the reliability and precision of its observation and the inherent predictability of its motion model. Simple in-animated objects for example may just follow physical equations of motion. Other autonomous agents on the other hand may change their intention and action unpredictably, while being harder to measure reliably due to their more complex shape, varying silhouette and changing backgrounds.

Each such distinction offers the possibility to apply approximations without losing too much precision in the estimation result. A more thorough analysis on the implication for those heterogeneous information sources and possible approximations can be found in [4]. In the implementation described in this paper, only a subset of those is employed.

The most relevant approximation in the context of this paper is the aggregation of measurements to build local short-term models of each observation type, thereby decreasing the uncertainty associated with the observed target and allowing to filter false positives. Once sufficiently recognized such a short-term model can be forwarded to the central estimation system as a meta-measurement and deleted from the temporary local model. The deletion of such models is important to preserve the independence assumption between consecutive measurements which is important for the Bayes filter concept. Insufficiently validated local hypotheses on the other hand can be pruned away without effecting a negative influence on the system's estimate. The short life span of those local models, e.g. below one second, prevents odometry errors to accumulate, but often allows the integration for example of a series of image processing results to obtain superior measurement quality.

2.3 Distributed Modeling

The sharing of information among a team of autonomous agents is especially desirable in cases where single robots have a very limited field of view and when occlusion frequently occurs. The distribution of information can be done with two conceptually different approaches. One approach can be classified as bottom-up and distributes measurements between robots, which are subsequently handled by common sensor fusion techniques. This is for example done in [6] and [7]. The top-down approach as applied in [8] and [9] consists of merging the individual robots' world maps. The prerequisite for such map merging is that all poses of participating robots need to be known, either in a consistent global coordinate frame or relative to each other. A common implementation in exploration scenarios is with uniquely identifiable robots which initiate map merging when observing each other, or when all robots are confidently localized in the global reference frame.

This latter approach would exclude poorly localized robots from map merging, however, those might also profit from the shared information, even specifically to resolve their poor localization in case of symmetries. If the measurements are distributed among robots, basically only each sending robot needs to be localized successfully in a global reference frame. An additional advantage is the computational and architectural simplicity of observation distribution compared to map merging, especially if observations are already aggregated in temporary local models as described in section 2.2, which in case of reliable localization can be distributed at the same time when used for the local integration into the global world model. Note that this approach does not guarantee a globally consistent model among all robots, since insufficiently localized robots do not send out information and therefore integrate more (but potentially also more unreliable) knowledge. The difference between the models of well localized robots however can be minimized by globally scheduling the exchange and integration phases of the observation distribution [4].

3 Implementation

The objective now is the realization of the demands specified in section 2, namely to model the robot's surrounding environment in one unified model using the information of a whole team of robots as input. This is hardly possible to implement as a real-time system on an embedded platform without applying measures to decrease the computational complexity. The presented approach consists of three stages, which will be covered in the following sections. The first stage handles the static map information to realize most (but not all) of the localization problem, and is based on an algorithm which can perform as a very efficient stand-alone localization [10]. Parallel to this runs a stage performing local percept aggregation according to the temporary short-term models described in section 2.2. Finally section 3.3 presents the integration of local and distributed perceptions into a consistent global world model.

In the following, each observation of a feature is represented by the two angles $z = (\alpha_1, \alpha_2)^T$ describing the direction in which the feature has been detected, and a third angle α_3 in case the feature has an identifiable orientation relative to the robot coordinate system. This is visualized in figure 1.

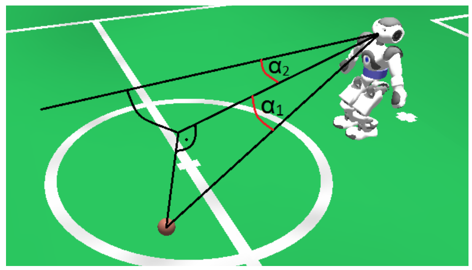


Fig. 1. Measurement of a feature expressed in horizon aligned observation angles.

3.1 Multi-model Kalman Localization

In contrary to the system described in [4], which is based on a particle filter localization, the approach presented here bases on a multi-hypothesis Unscented Kalman Filter (UKF) localization which has been presented in [10]. This utilizes an approach to Gaussian mixture filtering which combines the accuracy of the Kalman filter and the robustness of particle filters without sacrificing computational efficiency. This is done by pointing out similarities to particle filtering with an extremely low number of particles, and bypassing critical approximations in common Gaussian mixture algorithms.

Applying known techniques from both fields in a new combination results in a multi-hypotheses Kalman filter which is superior to common Kalman filters in its ability of fast re-localization in kidnapped robot scenarios and its representation of multi-modal belief distributions, and which outperforms particle filters in localization accuracy and computational efficiency. The output of this system is a set of robot pose hypotheses with corresponding covariances and likelihood estimations. If this is used as a separate localization module, the most likely hypothesis can be considered as the localization's result, and used as an input for behavior decisions or further planning.

To use this in the context of a unified world model it is necessary to keep track of the history of each hypothesis' origin for fusion and spawning of new hypotheses, and the change of the likelihoods among the set of hypotheses, which corresponds to a re-localization event for example with a kidnapped robot or after temporary localization loss caused by extreme odometry errors or collisions. Otherwise each estimate's change can be considered as a pre-filtered input for the global estimation system. This input bears the characteristics, on the one hand, of partially corrected odometry data, and on the other hand that one of buffered and pre-processed sensor data. In addition to this, integration of further information, including the communicated observations of other robots, can affect the pose estimates, so those changes need to be fed back into the localization module. The following sections will address the integration into the global model and the stochastic soundness of this.

3.2 Local Percept Aggregation

When building upon the UKF localization described above, the full state can not be factorized as in FastSLAM, but needs to be expressed as a joint probability function, as in the EKF-SLAM solution. The increase of estimation complexity by the high-dimensional state is countered by aggregation of some of the image processing results into temporary percept-buffers as motivated in section 2.2 with the aforementioned advantages.

This is applied to full extend to the dynamic features, i.e. the ball and other robots in a robot soccer scenario. Measurements of robots or the ball consist of two angles $z = (\alpha_1, \alpha_2)^T$ as described above. The estimated state consists of a 2-dimensional spacial component and a corresponding velocity component: $\mu = (x, y, v_x, v_y)^T$. For the spacial component of the state $\mu' = (x, y)^T$ and the height of the robot's camera r, equations 1 and 2 can be used to calculate the sensor model in form of the observation matrix H as in equation 3. Note that the velocity is not observable by processing a single camera image.

$$h_{\alpha_1}(\mu) = \operatorname{atan2}(r, |\mu'|) \tag{1}$$

$$h_{\alpha_2}(\mu) = \operatorname{atan2}(y, x) \tag{2}$$

$$H = \begin{pmatrix} -\frac{rx}{|\mu'|^3+r^2|\mu'|} & -\frac{ry}{|\mu'|^3+r^2|\mu'|} & 0 & 0 \\ -\frac{y}{x^2+y^2} & \frac{x}{x^2+y^2} & 0 & 0 \end{pmatrix} \tag{3}$$

For objects which are simply governed by the physical laws of motion, instead of being motorized or controlled, the motion model for the control update consists of a continuous motion slowed down by a friction factor $k = \frac{F_{friction}}{m}$ as the force generated by the friction divided by the mass of the object. Since the state is modeled in local coordinates, the robot's own motion, given by the translational and rotational odometry $(\delta_x, \delta_y, \delta_\theta)$, also transforms the local estimate. This results in the following time update for the velocity vector:

$$v_t = \begin{cases} \overbrace{\left(1 + \frac{k\Delta t}{|v_{t-1}|}\right) \Omega(-\delta_\theta)}^{=:V} \, v_{t-1} & \text{for } |v_{t-1}| \geq |k\Delta t| \\ \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} v_{t-1} & \text{else} \,. \end{cases} \tag{4}$$

where $\Omega(\alpha) = \begin{pmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{pmatrix}$ is the rotation around α. The full time update therefore predicts the state μ_{t-1} according to equation 5.

$$\overline{\mu}_t = g(\mu_{t-1}) = \begin{pmatrix} \Omega(-\delta_\theta) & \Delta t \Omega(-\delta_\theta) \\ 0 & V \end{pmatrix} \mu_{t-1} - \begin{pmatrix} \delta_x \\ \delta_y \\ 0 \\ 0 \end{pmatrix} \quad (5)$$

This results in the Jacobi matrix G for the process update as the partial derivatives of x,y,v_x and v_y at $(x_{t-1}, y_{t-1}, v_{x,t-1}, v_{y,t-1})$:

$$G = \begin{pmatrix} \Omega(-\delta_\theta) & \Delta t \Omega(-\delta_\theta) \\ 0 & M \end{pmatrix} \quad (6)$$

$$M = \begin{cases} \begin{pmatrix} \frac{\partial g_{v_x}}{\partial v_x} & \frac{\partial g_{v_x}}{\partial v_y} \\ \frac{\partial g_{v_y}}{\partial v_x} & \frac{\partial g_{v_y}}{\partial v_y} \end{pmatrix} & \text{if } |v_{t-1}| \geq |k \Delta t| \\ \Omega(-\delta_\theta) & \text{else} \end{cases} \quad (7)$$

with

$$\frac{\partial g_{v_x}}{\partial v_x} = \left(1 + \frac{k\ \Delta t}{|v|}\right) \cos(-\delta_\theta)\ - \frac{k\ \Delta t\ v_x\,(\cos(-\delta_\theta) v_x - \sin(-\delta_\theta) v_y)}{|v|^3} \quad (8)$$

$$\frac{\partial g_{v_x}}{\partial v_y} = -\left(1 + \frac{k\ \Delta t}{|v|}\right) \sin(-\delta_\theta)\ - \frac{k\ \Delta t\ v_y\,(\cos(-\delta_\theta) v_x - \sin(-\delta_\theta) v_y)}{|v|^3} \quad (9)$$

$$\frac{\partial g_{v_y}}{\partial v_x} = \left(1 + \frac{k\ \Delta t}{|v|}\right) \sin(-\delta_\theta)\ - \frac{k\ \Delta t\ v_x\,(\sin(-\delta_\theta) v_x + \cos(-\delta_\theta) v_y)}{|v|^3} \quad (10)$$

$$\frac{\partial g_{v_y}}{\partial v_y} = \left(1 + \frac{k\ \Delta t}{|v|}\right) \cos(-\delta_\theta)\ - \frac{k\ \Delta t\ v_y\,(\sin(-\delta_\theta) v_x + \cos(-\delta_\theta) v_y)}{|v|^3}\,. \quad (11)$$

Thus local models of dynamic objects in the robot's environment can be modeled using separate Kalman filters. In case of the unpredictability of the motion of autonomous robots it is possible to neglect the estimation of their velocity and apply high process noise instead.

The separate localization module described in section 3.1, in itself also a buffer integrating information from static, known world features into a localization belief model, is used analogically to those percept-buffers, but the state is not deleted periodically after forwarding the belief to the SLAM part of the algorithm. This localization reflects part of the SLAM state, and changes to this part of the SLAM state are fed back into the localization module's state. Thus the virtual localization measurements used to update the SLAM state are basically the innovation introduced by new static feature observations. Therefore those measurements are still conditionally independent from previous measurements given the current belief state, so the Markov assumption is not violated.

3.3 Local and Distributed Knowledge Integration

The state of the full model of the robot's environment consists of its own pose $p_0 = (p_{0,x}, p_{0,y}, p_{0,\theta})^T$, the poses of all cooperating robots $(p_i = (p_{i,x}, p_{i,y}, p_{i,\theta})^T$

with $i \in 1, ..., n$), and the states of the dynamic objects. While only a small subset of cooperating robots or other elements are observed at the same time and modeled according to section 3.2 in each time interval, they remain part of the full model also during time intervals where these are not observed. It is possible to dynamically shrink or expand the state vector if new unknown robots are observed. Alternatively a separate mechanism could keep track of active and inactive slots in the state vector by using time-to-live counters. This latter approach has been chosen here to prevent frequent rescaling of both the state vector and its covariance matrix.

The integration of the locally accumulated and the distributed information into the model will be done in the process and sensor update. The own pose and those of cooperating robots can be updated with the pose changes propagated from the individual localization modules relative to the pose used for the last update. The ball is updated using a motion model similar to the one in equation 5, but without the odometry related rotations due to the local coordinate system. Other autonomous agents can either be updated according to the latest velocity estimations, or just using an identity and appropriately high process noise following the reasoning proposed in section 2.2.

The sensor update consists of two different kinds of observations. If a robot, either the local robot itself or any of the communicating robots in the team, has made observations of static world elements which have been used to update the separate localization estimate in the first stage (cp. section 3.1), then this absolute pose estimate is used as a direct measurement of the corresponding pose in the state vector, i.e. the measurement Jacobian is an identity in the corresponding submatrix.

The other case is the observation of a dynamic feature by one of the robots in the team. If the observed dynamic feature is a robot (without further identified characteristics such as team markers etc.), this dynamic object may either be any of other robots in the team, or one of a number of non-cooperating other robots in the environment. In this case, the maximum likelihood correspondence will be chosen to be updated, or a new model will be inserted or activated if the other choices are too unlikely. The corresponding expected observation is in a robot-relative euclidean coordinate system, since this is the format of the local models distributed as aggregated percepts. It is expressed as a function of the observed object's model $(m_x, m_y, m_{v_x}, m_{v_y})$ and its observer's pose p_i, with $i = 0$ for local observations and $i \in 1, ..., n$ communicated ones, which are otherwise not distinguished any further.

The observation model is given by equations 12 and 13

$$h_{m_x, m_y}(p_i) = \Omega(-p_{i,\theta}) \left[(m_x, m_y)^T - (p_{i,x}, p_{i,y})^T \right] \tag{12}$$

$$h_{m_{v_x}, m_{v_y}}(p_i) = \Omega(-p_{i,\theta}) \left(m_{v_x}, m_{v_y} \right)^T \tag{13}$$

from which the corresponding entries in the measurement Jacobian can be calculated as in equation 14, with c_θ and s_θ short for $\cos p_{i,\theta}$ and $\sin p_{i,\theta}$, respectively.

$$
\begin{pmatrix}
\frac{\partial h_{m_x}}{\partial m_x} & \frac{\partial h_{m_x}}{\partial m_y} & \frac{\partial h_{m_x}}{\partial m_{v_x}} & \frac{\partial h_{m_x}}{\partial m_{v_y}} & \frac{\partial h_{m_x}}{\partial p_{i,x}} & \frac{\partial h_{m_x}}{\partial p_{i,y}} & \frac{\partial h_{m_x}}{\partial p_{i,\theta}} \\
\frac{\partial h_{m_y}}{\partial m_x} & \frac{\partial h_{m_y}}{\partial m_y} & \frac{\partial h_{m_y}}{\partial m_{v_x}} & \frac{\partial h_{m_y}}{\partial m_{v_y}} & \frac{\partial h_{m_y}}{\partial p_{i,x}} & \frac{\partial h_{m_y}}{\partial p_{i,y}} & \frac{\partial h_{m_y}}{\partial p_{i,\theta}} \\
\frac{\partial h_{m_{v_x}}}{\partial m_x} & \frac{\partial h_{m_{v_x}}}{\partial m_y} & \frac{\partial h_{m_{v_x}}}{\partial m_{v_x}} & \frac{\partial h_{m_{v_x}}}{\partial m_{v_y}} & \frac{\partial h_{m_{v_x}}}{\partial p_{i,x}} & \frac{\partial h_{m_{v_x}}}{\partial p_{i,y}} & \frac{\partial h_{m_{v_x}}}{\partial p_{i,\theta}} \\
\frac{\partial h_{m_{v_y}}}{\partial m_x} & \frac{\partial h_{m_{v_y}}}{\partial m_y} & \frac{\partial h_{m_{v_y}}}{\partial m_{v_x}} & \frac{\partial h_{m_{v_y}}}{\partial m_{v_y}} & \frac{\partial h_{m_{v_y}}}{\partial p_{i,x}} & \frac{\partial h_{m_{v_y}}}{\partial p_{i,y}} & \frac{\partial h_{m_{v_y}}}{\partial p_{i,\theta}}
\end{pmatrix}
$$

$$
= \begin{pmatrix}
c_\theta & s_\theta & 0 & 0 & -c_\theta & -s_\theta & -(m_x - p_{i,x}) \cdot s_\theta + (m_y - p_{i,y}) \cdot c_\theta \\
-s_\theta & c_\theta & 0 & 0 & s_\theta & -c_\theta & -(m_x - p_{i,x}) \cdot c_\theta - (m_y - p_{i,y}) \cdot s_\theta \\
0 & 0 & c_\theta & s_\theta & 0 & 0 & -m_{v_x} \cdot s_\theta + m_{v_y} \cdot c_\theta \\
0 & 0 & -s_\theta & c_\theta & 0 & 0 & -m_{v_x} \cdot c_\theta - m_{v_y} \cdot s_\theta
\end{pmatrix} \tag{14}
$$

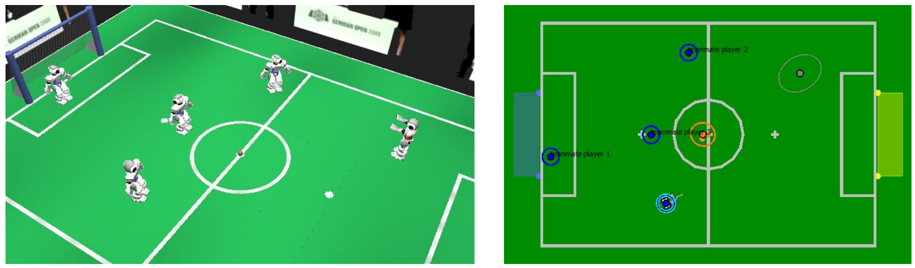
Re-localization events can be handled by resetting the corresponding state variables and removing the covariances, i.e. setting all entries in the covariance matrix in the rows and columns to zero. If such a previous mis-localization by a team member resulted in modeled false positives, those will stay as isolated features in the state for some time and will be deleted or inactivated after a certain time without observation. This serves as a self-repair routine to remove clutter from the environmental model, and to prevent the growth of the state by the accumulation of models of such elements. The same is done if two models of unknown features are decided to correspond to the same origin after a series of observations, so that the information needs to be fused into the first model and the seconds needs to be deactivated. Alternatively it would be possible to keep multiple environment models for each localization hypothesis, as done in [4].

4 Evaluation

The modeling process is complex and incorporates a multitude of different information, so that a step by step illustration of the working principle is not practical. To evaluate the presented approach, a simulated situation first illustrates the theoretical possibilities and the qualitative effect in section 4.1, followed by a quantitative analysis in soccer games using experiments with real robots in section 4.2. Both setups use an SPL scenario as specified by the 2011 rules.

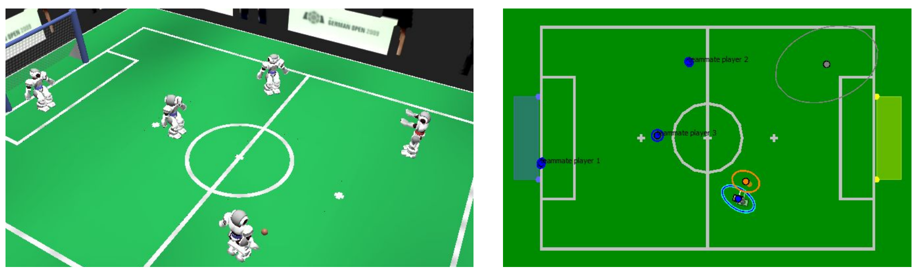
4.1 Qualitative Demonstration

Figure 2 illustrates a simple scenario in a simulated environment. The robots in a team share their information for distributed cooperative modeling. Figure 2(b) shows the resulting model with 2D covariance ellipses extracted from the full state. In the following, one robot looks down and does not see any static field features any more, and both he and the ball are teleported to another location on the field (see figure 3). The use of distributed percepts and the modeling of the own pose together with the ones of other robots and the ball position and velocity allows the robot to not only correct its position, but also its orientation.



(a) Setup of the robots on the field. (b) World model generated from local and distributed information.

Fig. 2. Scenario with a team of robots looking around and sharing perception information to cooperatively model their environment.



(a) Scenario after teleportation of ball and downwards-looking robot. (b) World model generated from local and distributed information.

Fig. 3. Following the situation in figure 2, one robot looks down and only sees the ball but no landmarks, and he and the ball are teleported. The shared information however still allows for a correction of both position and orientation of the robot.

This simple experiment shows the potential usefulness of such a combined modeling of a robot's dynamic environment and its pose in it. RoboCup SPL games contain periods where robots are chasing the ball, approaching it for precise positioning to shoot at the goal, or even dribbling it. During those periods odometry errors are integrated into the robot's localization if not countered by frequently looking up at static field features to correct the robot's pose estimation. If looking at the ball also allows the correction of those odometry errors, especially the orientation, this is expected to be a clear advantage.

4.2 Quantitative Performance Evaluation

The artificial situation created in the previous section just serves as an example of how localization benefits may be gained. To allow a quantitative evaluation of the approach's performance, the perceptions of a robot have been recorded during normal game situations with real robots on a regular SPL field. Those perceptions include the proprioception, i.e. odometry, orientation and joint angle information, exteroception, i.e. perceptions of objects by means of image processing, as well as the distributed local models of other cooperating robots running the same code, and ground truth information provided by a camera system mounted above the field.

This set of input information is then processed by two different module configurations. One is the configuration described in section 3. The second uses the same localization but a simpler module for cooperative tracking of dynamic objects without any feedback into the localization, and has been used to win the second place at RoboCup 2011. This experiment is not set up to show that the localization works, since both solutions are based on the same competitive solution for the localization problem with all features described in [10], but to evaluate the additional benefit gained by unified modeling of the full state.

A first evaluation of several recorded situations did not show any conclusive results, meaning the positive and negative effects of the full state modeling equaled out most of the time, in a low percentage of cases the full system even showed a slightly decreased localization quality. Closer evaluation showed that the currently used visual robot recognition provides too much uncertainty or even uncorrected systematical errors, such as in the distance estimation, to be beneficial for the localization.

A second configuration of the system, which ignores the robot perceptions for the modeling of the robot's own position, but still uses the much more precise ball perceptions, showed the expected results. As can be seen in the representative extract visualized in figure 4, the proposed system provides beneficial information for the robot's own localization most of the time. A direct comparison of the localization quality of both systems shows that the robot pose translation errors for the full system model are below 25 cm in 83% of the time, and only 72% of the time for the unassisted underlying localization module, and the average errors over the whole experiment are 166 mm compared to 213 mm. However, note that with this second configuration of the system, in the teleportation experiment the robot's orientation could not be recovered as easily as described in section 4.1.

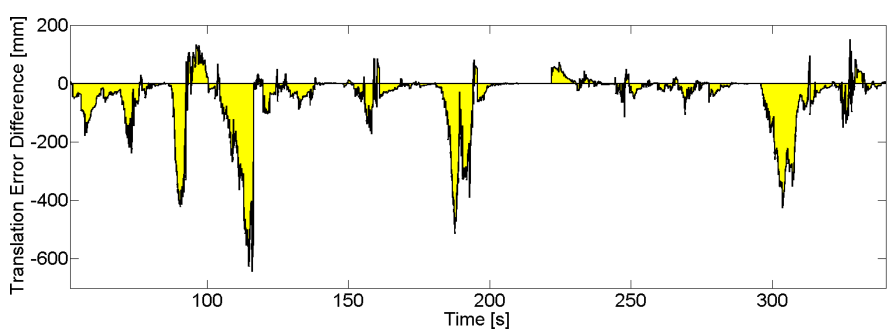


Fig. 4. Difference of translation errors of the two described systems. Negative values mean larger errors of the unassisted localization compared to modeling the full state.

5 Conclusion

This paper presents the advantages of modeling the full environment state estimation as compared to only localizing in said environment. A competitive stand-alone localization module is extended to perform as a full state model, and the additional gain in localization performance is evaluated both in a simulated situation as well as in several real world experiments with multiple robots and ground truth provided by an external camera system. While the robot perception in the current vision system is not good enough to benefit from using temporary opponent models as additional features for localization, usage of the ball as a dynamic feature significantly improves the localization quality.

An additional advantage of estimating the full state in a cooperative modeling approach is the existence of a single model which contains all information in a globally consistent way. This renders the switching between local tracking of the ball and a global team ball model obsolete, for example, and therefore simplifies behavior specification.

References

1. Thrun, S., Burgard, W., Fox, D.: Probabilistic Robotics (Intelligent Robotics and Autonomous Agents). The MIT Press (2005)
2. Hähnel, D., Schulz, D., Burgard, W.: Map building with mobile robots in populated environments. In: Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). (2002)
3. Schulz, D., Burgard, W., Fox, D., Cremers, A.B.: Tracking multiple moving objects with a mobile robot. Computer Vision and Pattern Recognition, IEEE Computer Society Conference on **1** (2001) 371
4. Czarnetzki, S., Rohde, C.: Handling heterogeneous information sources for multi-robot sensor fusion. In: Proceedings of the 2010 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI 2010), Salt Lake City, Utah (September 2010) 133 – 138
5. Kümmerle, R., Steder, B., Dornhege, C., Kleiner, A., Grisetti, G., Burgard, W.: Large scale graph-based SLAM using aerial images as prior information. In: Proceedings of Robotics: Science and Systems (RSS), Seattle, WA, USA (June 2009)
6. Stroupe, A., Matrin, M., Balch, T.: Distributed sensor fusion for object position estimation by multi-robot systems. In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), 2001. (2001)
7. Dietl, M., Gutmann, J.S., Nebel, B.: Cooperative Sensing in Dynamic Environments. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). (2001) 1706–1713
8. Howard, A.: Multi-robot simultaneous localization and mapping using particle filters. The International Journal of Robotics Research **25**(12) (2006) 1243–1256
9. Zhou, X.S., Roumeliotis, S.I.: Multi-robot SLAM with unknown initial correspondence: The robot rendezvous case. In: Proceedings of IEEE International Conference on Intelligent Robots and Systems (IROS), Beijing, China (2006) 1785–1792
10. Jochmann, G., Kerner, S., Tasse, S., Urbann, O.: Efficient multi-hypotheses unscented kalman filtering for robust localization. In Röfer, T., Mayer, N.M., Savage, J., Saranli, U., eds.: RoboCup 2011: Robot Soccer World Cup XV. Lecture Notes in Computer Science. Springer Berlin / Heidelberg (2012) to appear