Improving Robot Self-Localization using Landmarks' Poses Tracking and Odometry Error Estimation*

Pablo Guerrero and Javier Ruiz-del-Solar

Department of Electrical Engineering, Universidad de Chile
{pguerrer,jruizd}@ing.uchile.cl

Abstract. In this article the classical self-localization approach is improved by estimating, independently from the robot's pose, the robot's odometric error and the landmarks' poses. This allows using, in addition to fixed landmarks, dynamic landmarks such as temporally local objects (mobile objects) and spatially local objects (view-dependent objects or textures), for estimating the odometric error, and therefore improving the robot's localization. Moreover, the estimation or tracking of the fixed-landmarks' poses allows the robot to accomplish successfully certain tasks, even when having high uncertainty in its localization estimation (e.g. determining the goal position in a soccer environment without directly seeing the goal and with high localization uncertainty). Furthermore, the estimation of the fixed-landmarks' pose allows having global measures of the robot's localization accuracy, by comparing the real map, given by the real (a priori known) position of the fixed-landmarks, with the estimated map, given by the estimated position of these landmarks. Based on this new approach we propose an improved self-localization system for AIBO robots playing in a RoboCup soccer environment, where the odometric error estimation is implemented using Particle Filters, and the robot's and landmarks' poses are estimated using Extended Kalman Filters. Preliminary results of the system's operation are presented.

1 Introduction

Localization is a key feature of a mobile robotic system, which has been deeply investigated over the last years. Commonly a localization module is expected to filter two sources of error: (i) observational errors that are produced by the imperfections of the sensors and their models, and (ii) odometric errors that are produced by flaws in the modeling of the actuators and by events that are very difficult to model as slipping and collisions. It is not the aim of this paper to compare or to analyze different localization methods -as it is a very largely discussed matter- but to discuss how to improve the localization process.

Existent localization approaches filter simultaneously both sources of error, observational and odometric, making use of what we call *global information*, perceptions of objects with fixed and known global pose. However, we believe that,

* This research was partially supported by FONDECYT (Chile) under Project Number 1061158.

in addition to the global information, additional sources of information, what we call *local information*, can be exploited by localization methods. We use the word "local" in its temporal and spatial meanings. Spatially local information corresponds to information that is only useful in a reduced region of the space, while temporally local information corresponds to information only valid in a short period of time. Temporally local information corresponds mainly to mobile objects, whose poses' estimates are valid for a short period of time. Spatially local information corresponds mainly to view-dependent objects or textures, whose perceptions are valid in a reduced neighborhood. The spatially restricted utility may be at least due to three reasons: (i) the object is only observable from a restricted region of the space, for example a design in the floor or a visual feature or detail only perceptible from close positions, (ii) its appearance changes from different points of view (most of the natural objects do not have a radial symmetry and also non isotropic light may make them appear different from different points of view), and/or (iii) several identical -or difficult to distinguish- objects are present in different places in the space, which could easily lead to confusion, for example, a tree in a forest, a tile or texture in the floor, or a chair in a classroom. SLAM approaches can deal with objects locally observable or with non-symmetric appearance by creating and maintaining a pose estimate for each of them, or of their different appearances treated as different objects. However, this could lead a system to maintain millions of estimates, which is computationally infeasible and practically senseless. Nevertheless, it is possible to think in a SLAM-like approach that maintains locally relevant information with the purpose of estimating the odometric error. We believe such an approach is more biologically inspired. For instance, humans are able to correct their odometry even when they have no knowledge of the environment.

In this context we propose improving the classical self-localization approach by estimating, independently from the robot's pose, the robot's odometric error and the landmarks' poses. This allows using, in addition to fixed landmarks, dynamic landmarks such as temporally local objects (mobile objects) and spatially local objects (view-dependent objects or textures), for estimating the odometric error, and therefore improving the robot's localization. Moreover, the estimation or tracking of the fixed-landmarks' poses allows the robot to carry out certain tasks, even when having high uncertainty in the localization estimation. This is especially valuable when performing attention demanding tasks, like pursuing a ball, which forbid the use of active vision in order to get more (standard) landmarks' perceptions. Another nice feature of the proposed system is that the robot is able to correct its odometry even when it is totally lost. The latter ability may be useful in several situations as for example, when shooting the ball to a recently seen goal, by correcting the relative robot's pose estimation with only observations of the ball. In this sense, we believe our approach also goes in the direction towards performing tasks with much less use of global localization, as certainly humans do.

Furthermore, the estimation of the fixed-landmarks' pose allows having global measures of the robot's localization accuracy, by comparing the real map, given by the real (a priori known) position of the fixed-landmarks, with the estimated map, given by the estimated position of these landmarks.

Based on the described new self-localization approach, we propose an improved self-localization system for AIBO robots playing in a RoboCup soccer environment, that implements odometric error estimation using Particle Filters, and robot's and landmarks' poses estimation using Extended Kalman Filters.

How and when to select spatially local observations as valid landmarks is a topic not addressed in this article. In the current implementation we consider temporally local observations, mobile objects, such as the ball and robot players in a soccer environment.

This paper is organized as follows. In section 2 is presented some related work. The improved self-localization approach is described in section 3. In section 4 some features of the proposed approach are discussed. In section 5, preliminary results are presented. Finally, in section 6 some conclusions and future work are given.

2 Related work

Standard Bayesian-based robot self-localization fuses odometric information with perceptual information coming from different sensors. Thus, odometry is employed for predicting the next robot pose state using a cinematic model of the robot, while perceptual information from landmarks is employed for correcting this prediction using an observational model. For implementing these two steps, the most employed Bayesian filters are Kalman [6] and Particle Filters [3][5]. Kalman Filtering is a very well-known technique for parameter estimation, its main drawback being the linearity and Gaussianity assumptions. EKF extends the Kalman Filtering idea by linearizing the measurement and plant (in this case the robot) models, but still has the assumption of Gaussianity [4]. On the other hand, particle filters overcome the drawbacks of assuming linearity and Gaussianity, by implementing a "factored sampling" of the processes' conditional densities [5]. Particle Filtering is very popular in the Computer Vision community where the most employed implementation is called *Condensation* [5], while in the mobile robotics community the most used implementation is the *Monte Carlo Localization – MCL* algorithm. However, particle filters have an important drawback, their performance depends strongly on the number of particles. In specific applications as robot soccer, in which the computational resources are limited, the number of employed particles may not be very high (normally between 50 and 200), and therefore particle filters do not clearly outperforms EKF. As a fact, in the RoboCup soccer leagues successful teams use either EKF, MCL, or mixtures of both (see for example [8] or [9]).

Nevertheless, it is not our intention to analyze or to compare different Bayesian filters and their application to the robot self-localization problem, but to improve the standard Bayesian-based robot self-localization by including new independent stages for estimating the robot's odometric error and the landmarks' poses. To the best of our knowledge this idea is novel, and has not being implemented before in robot localization systems. Although decoupling the odometry estimation from the landmark position estimation has been proposed in the SLAM literature [10], there is a strong implicit assumption in these works, which is that all the landmarks will remain static forever. In some SLAM approaches detected objects are tracked and characterized as mobile or static [11]. However the mobile object's information is not used for estimating the odometric error.

In visual odometry approaches (see for example [12][13]) local visual features (e.g. Harris or SIFT features) are employed for estimating the robot relative movements (the odometry) by detecting and matching the features between consecutive frames. The main differences with our approach are: (1) In our approach the estimated odometry is used in the robot localization process, and also for updating the high-level tracking of landmarks. Traditional visual odometry approaches do not include high-level tracking of landmarks, therefore the use of the estimated odometry is much simpler; (2) Visual odometry approaches use local features, while in our case high-level fixed or moving landmarks are employed (e.g. a ball or a goal in a soccer environment). We believe that using high-level landmarks is more robust because: (i) local features cannot be detected in any environment or moment (e.g. in a robot soccer carpet no local features can be detected), and (ii) when analyzing images corresponding to real-world environments the process of matching local features can produce a larger number of false matches (due to shadows, highlights, symmetry problems, too many detected features, etc.) than the one of matching high-level feature.

3 Proposed Self-Localization using Landmarks' Tracking and Odometry Error Estimation

As already mentioned the basic idea of the proposed approach is to estimate, independently from the robot's pose, the robot's odometric error and the landmarks' poses. For achieving this two new processes are included High-Level Tracking (*HL-Tracking* module) and Odometry Error Estimation (*OEE* module). As can be observed in figure 1, the operation of all process is tightly interconnected. First, in the *HL-Tracking* module the pose of the observed landmark ($\mathbf{x}_{l_*,k}^{--}$), either static or mobile, is *early* predicted using the odometry information ($\mathbf{u}_{k-1}$). Then, the odometric error ($\mathbf{e}_k$) is estimated in the *OEE* module using the information of current observations (coming from *Vision*) ($\mathbf{z}_k$), and the corresponding landmark's early estimated pose. Afterwards, in the *HL-Tracking* module the estimated odometric error is used for estimating the new landmarks' poses ($\{\mathbf{x}_{l_i,k}\}$). Finally, the corrected odometry and the landmarks' poses are employed for estimating the robot's localization ($\mathbf{x}_{R,k}$).

In the next sections the operation of all modules is described in detail for the case of a RoboCup four-legged environment using AIBO robots. The pseudo code and equations are detailed in tables 1-3.

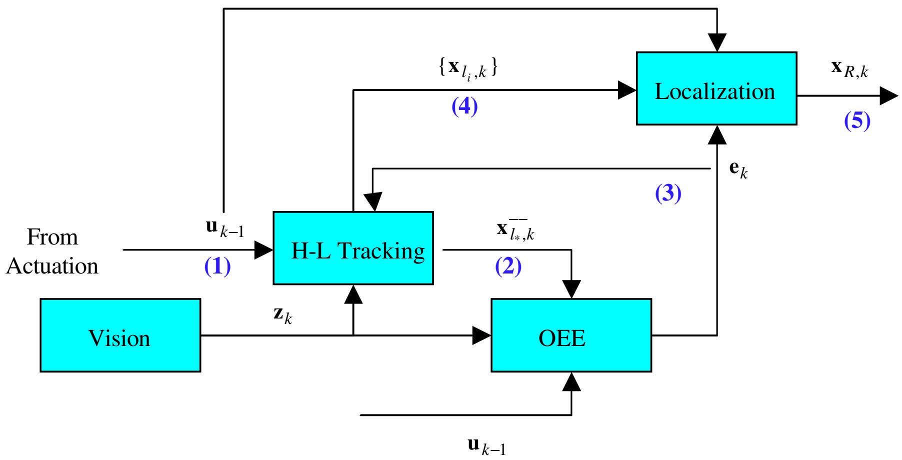

Fig. 1. Block diagram of the system. Two stages are added between vision and localization: HL-Tracking and Odometry Error Estimation.

3.1 Vision

Our vision system (RoboCup four-legged league scenario) is based on color segmentation of the images, and rule based perceptors for the relevant objects (ball, robots, beacons and goals) (see detailed description in [1]). The vision system also includes a recently proposed context filter, which takes into account the coherence between current and past detections, as well as scene and situation contexts, to filter incoherent detections [2].

3.2 HL-Tracking

In this module the state of every detected and coherent object/landmark ($\mathbf{x}_{l_i,k}$) is tracked (estimated). For the fixed objects, the state corresponds to their 2D pose, relative to the robot, for the mobile ones, a relative velocity is also added (see figure 2). The pose of any object includes a 2D position, and may include a relative orientation, if this is distinguishable (for objects with radial symmetry as ball and beacons it is not possible to notice their orientation).

The current implementation of the HL-Tracking stage consists of one independent EKF for each object. The prediction of each EKF has two stages: (i) early prediction (step (1) in pseudo code shown in table1), where the pose of the l_* landmark associated with the current observation $\mathbf{z}_k$, is predicted using the last executed odometry $\mathbf{u}_{k-1}$, and (ii) a standard prediction stage (step (6) in pseudo code), where the poses of all landmarks ($\{\mathbf{x}_{l_i,k}\}$) is predicted using the corrected odometry ($\mathbf{u}_{k-1} + \mathbf{e}_k$). The correction stage is standard and considers only the observed landmark l_* (step (7) in pseudo code). In the case of mobile objects, the correction that the filter takes is standard and very straightforward, since the predicted observation may be extracted directly from the state -the observation model Jacobian H is equal to the identity or some submatrix-.

For the system to be able to quickly detect and recover from kidnaps, all tracked estimates of objects' poses in HL-Tracking has a smoothed object coherence indicator (see [2] for details).

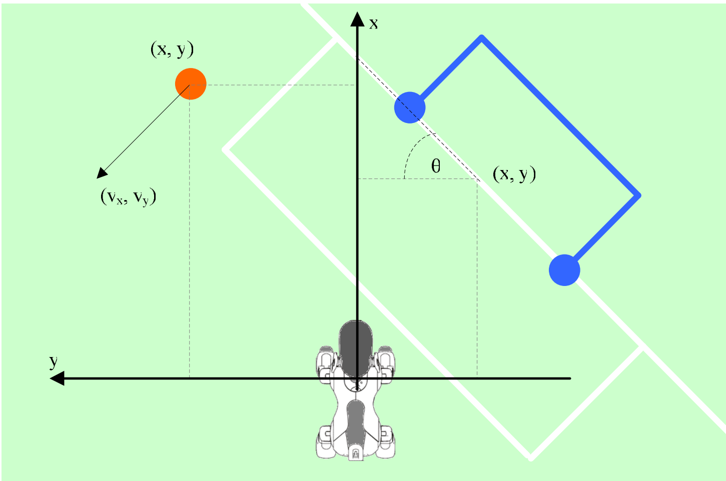

Fig. 2. Tracked objects and their state in the HL-Tracking stage. Static objects are represented by their relative pose, which includes a position and may include a relative orientation (blue goal). For mobile objects, a relative velocity is also added (orange ball).

3.3 Odometry Error Estimation

The odometry error estimation (*OEE*) stage is implemented using a particle filter, as in MCL, but in this case each particle represents a hypothesis for the accumulated odometric error (instead of the global pose of the robot, as in MCL). Consequently, the state of each particle is a pose (x, y, θ) relative to a coordinate system centered in an odometry error-free pose. Then, the particles are drawn over the same coordinate system shown in figure 2.

In the sampling stage of the OEE (step (3) in pseudo code), it is considered that the expected odometric error is zero, thus, we only add noise covariance to the particles, coming from the standard odometry. Given any odometry $\mathbf{u}_{k-1}$, an a priori odometry error covariance $\mathbf{Q}_{k-1}$ is used to scatter the particles. The weighting stage consists in the calculation of weights for each particle (step (4) in pseudo code). A particle will have a higher weight when it better explains the difference between the observed and estimated poses of the observed objects. Finally, in the Resampling Stage (step (2) in pseudo code), particles are resampled according to their weights. A particle is copied a number of times that has an expected value proportional to its weight. As Resampling is the first stage in our implementation, it considers the weights calculated in the previous iteration of the system.

Every time an observation arrives, the system executes these three steps, and finally the estimated odometric error ($\mathbf{e}_k$) and its covariance ($\mathbf{Q}_{\mathbf{e},k}$) are statistically calculated over the particles (step (5) in pseudo code), and used as additional predictive inputs for the HL-Tracking and Localization stages. Finally, the odometric error estimate is subtracted from each particle, to set the new odometry error estimate to zero.

3.4 Localization

We have implementations of standard robot's localization modules based on EKF, MCL, and mixtures of them [1]. However, for the proposed approach, we have implemented the robot's localization using a standard EKF filter (see pseudo code in table 2). The main new features are: (i) the corrected odometry ($\mathbf{u}_{k-1} + \mathbf{e}_k$) is used for predicting the new robot's pose (step (8) in the pseudo code), and (ii) the filtered relative poses of all landmarks, fixed and mobile, are employed as the filter's observations in the correction stage (step (9) and (10) in pseudo code).

4 Discussion

4.1 Is it good for a robot to be egocentric?

A clear difference between the proposed approach and most of the existent approaches for localization is that, in this one, all the analysis is made in reference to the robot instead of making it from a global point of view. One could argue that all the previous formulation could be transported to a global approach by representing all the local information in a global coordinate system. We believe that being egocentric (taking a self-centered coordinate system for most of the calculations) is a good decision because: (i) many (may be most) of the tasks a robot must perform can be executed with only local information, for example, a robot does not need to know its global pose neither the global pose of the ball to approach to it, and (ii) even high level tasks that need global information normally result in low-level tasks that can be performed locally.

Table 1. HL-Tracking and OEE pseudo code and equations. See definition in table 3.

(1) High-Level Tracking Early Prediction Stage // Early prediction using odometry

$\mathbf{x}^{--}_{l_*,k} = f_l(\mathbf{x}_{l_*,k-1}, \mathbf{u}_{k-1}, 0)$ // l_* the landmark associated with the current observation $\mathbf{z}_k$, and $\mathbf{u}_{k-1}$ the last executed odometry

(2) Odometry Error Resampling Stage

Resample $\left\{\mathbf{x}'_{p_j,k}\right\}$ according to $\left\{\mathbf{x}_{p_j,k-1}, \omega_{p_j,k-1}\right\}$, $j = 1,...,T$ //T: total number of particles

(3) Odometry Error Sampling Stage

$\mathbf{Q}_{k-1} = \mathbf{Q}(\mathbf{u}_{k-1})$ // a priori odometry error covariance

For each particle $p_j, j = 1,...,T$ do:

$\mathbf{u}_{p_j,k} \sim N(0, \mathbf{Q}_{k-1})$ //Normal distribution with zero mean and $\mathbf{Q}_{k-1}$ covariance

$\mathbf{x}_{p_j,k} = f_p(\mathbf{x}'_{p_j,k}, \mathbf{u}_{p_j,k}, 0)$

(4) Odometry Error Weighting Stage

For each particle $p_j, j = 1,...,T$ do:

$\mathbf{v}_{p_j,k} = \mathbf{z}_k - \mathbf{h}_{l_*}(f_l(\mathbf{x}^{--}_{l_*,k}, \mathbf{u}(\mathbf{x}_{p_j,k}), 0), 0)$ // $\mathbf{z}_k$ the current observation, and l_* the corresponding landmark

$$\tilde{\omega}_{p_j,k} = e^{-\frac{\mathbf{v}^T_{p_j,k}\mathbf{R}^{-1}_{l_j,k}\mathbf{v}_{p_j,k}}{2}}$$

$$\omega_{p_j,k} = \frac{\tilde{\omega}_{p_j,k}}{\sum_n \tilde{\omega}_{p_n,k}}$$ // weights normalization

(5) Odometry Error Statistics Calculation

$\mathbf{e}_k = \sum_j \omega_{p_j,k}\mathbf{x}_{p_j,k}$ // estimated odometry error

$\mathbf{Q}_{\mathbf{e},k} = \sum_j \omega_{p_j,k}\mathbf{x}_{p_j,k}\mathbf{x}^T_{p_j,k}$ // estimated odometry error covariance

For each particle $p_j, j = 1,...,T$ do:

$\mathbf{x}_{p_j,k} = \mathbf{x}_{p_j,k} - \mathbf{e}_k$

(6) High-Level Tracking Odometry Prediction Stage

For each landmark $l_i, i = 1,...,L$ do: // The new poses of all landmarks are predicted

$\mathbf{x}^-_{l_i,k} = f_l(\mathbf{x}_{l_i,k-1}, (\mathbf{u}_{k-1} + \mathbf{e}_k), 0)$ // Prediction using the corrected odometry

$\mathbf{P}^-_{l_i,k} = \mathbf{A}_{l,k}\mathbf{P}_{l_i,k-1}\mathbf{A}^T_{l,k} + \mathbf{W}_{l_i,k}\mathbf{Q}_{\mathbf{e},k}\mathbf{W}_{l_i,k}$

(7) High-Level Tracking Correction Stage // Only the observed-landmark's pose is corrected

$\mathbf{v}_{v,k} = \mathbf{z}_k - \mathbf{h}_{l_*}(x^-_{l_*,k}, 0)$ // $\mathbf{z}_k$ the current observation, and l_* the corresponding landmark

$\mathbf{M}_{l_*,k} = \mathbf{H}_{l_*,k}\mathbf{P}^-_{l_*,k}\mathbf{H}^T_{l_*,k} + \mathbf{V}_{l_*,k}\mathbf{R}_{l_*,k}\mathbf{V}^T_{l_*,k}$

$\mathbf{K}_{l_*,k} = \mathbf{P}^-_{l_*,k}\mathbf{H}^T_{l_*,k}\mathbf{M}^{-1}_{l_*,k}$

$\mathbf{x}_{l_*,k} = \mathbf{x}^-_{l_*,k} + \mathbf{K}_{l_*,k}\mathbf{v}_{v,k}$

$\mathbf{P}_{l_*,k} = (\mathbf{I} - \mathbf{K}_{l_*,k}\mathbf{H}_{l_*,k})\mathbf{P}^-_{l_*,k}$

Table 2. EKF Localization pseudo code and equations. See definition in table 3.

(8) Localization Prediction Stage // Robot pose prediction using the corrected odometry

$$\mathbf{x}^-_{R,k} = f_R(\mathbf{x}_{R,k-1}, (\mathbf{u}_{k-1} + \mathbf{e}_k), 0)$$

$$\mathbf{P}^-_{R,k} = \mathbf{A}_{R,k}\mathbf{P}_{R,k-1}\mathbf{A}^T_{R,k} + \mathbf{W}_{R,k}\mathbf{Q}_{e,k}\mathbf{W}^T_{R,k}$$

(9) Localization Data Association Stage

$$\mathbf{v}_{l,k} = \varnothing$$

For each landmark $l_i, i = 1,...,L$ do: //Landmarks are filtered out using an innovation threshold g

$$\mathbf{v}_{l_i,k} = \mathbf{x}_{l_i,k} - h_{R,l_i}(\mathbf{x}^-_R, 0)$$

$$\mathbf{M}_{R,l_i,k} = \mathbf{H}_{R,l_i,k}\mathbf{P}^-_R\mathbf{H}^T_{R,l_i,k} + \mathbf{V}_{R,l_i,k}\mathbf{P}_{l_i,k}\mathbf{V}^T_{R,l_i,k}$$

$$if\,(\mathbf{v}_{l_i,k}\mathbf{M}_{R,l_i,k}\mathbf{v}^T_{l_i,k} \le g^2)\;\; then\;\; \mathbf{v}_{l,k} = \mathbf{v}_{l,k} \cup \mathbf{v}_{l_i,k}$$

(10) Localization Correction Stage // Robot pose correction

$$\mathbf{K}_{R,k} = \mathbf{P}^-_{R,k}\mathbf{H}^T_{R,k}\mathbf{M}^{-1}_{R,k}$$

$$\mathbf{x}_{R,k} = \mathbf{x}^-_{R,k} + \mathbf{K}_{R,k}\mathbf{v}_{l,k}$$

$$\mathbf{P}_{R,k} = (\mathbf{I} - \mathbf{K}_{R,k}\mathbf{H}_{R,k})\mathbf{P}^-_{R,k}$$

Table 3. Variables, matrices and functions definitions.

Variable/Matrix	**Definition**
$\mathbf{x}_{l_i,k}$, $\mathbf{P}_{l_i,k}$	Landmark l_i state vector and covariance matrix.
$\mathbf{x}_{R,k}$, $\mathbf{P}_{R,k}$	Robot state vector and covariance matrix.
$\mathbf{Q}_k$	A priori odometry error covariance
f_l, f_R	Landmarks and robot process models.
$\mathbf{u}_{k-1}$	Robot odometry.
$\mathbf{e}_k$ and $\mathbf{Q}_{\mathbf{e},k}$	Estimated odometric error and its covariance.
$\mathbf{x}_{p_j,k}$ and $\omega_{p_j,k}$	Position and weigh of particle p_j (odometry error estimation)
f_p	Cinematic model of the particles (odometry error estimation)
$\mathbf{A}_{R,k}$ / $\mathbf{W}_{R,k}$	The Jacobian matrix of the partial derivatives of f_R with respect to the state vector and process noise, respectively.
$\mathbf{A}_{l,k}$ / $\mathbf{W}_{l,k}$	The Jacobian matrix of the partial derivatives of f_l with respect to the state vector and process noise, respectively.
$\mathbf{H}_{R,k}$ / $\mathbf{V}_{R,k}$	The Jacobian matrix of the partial derivatives of h_R with respect to the state vector and observational noise, respectively.
$\mathbf{H}_{R,l_i,k}$ / $\mathbf{V}_{R,l_i,k}$	The Jacobian submatrix, corresponding to the landmark l_i, of the partial derivatives of h_R with respect to the state vector and observational noise, respectively.
$\mathbf{H}_{l_i,k}$ / $\mathbf{V}_{l_i,k}$	The Jacobian matrix of the partial derivatives of h_{l_i} with respect to the state vector and observational noise, respectively.
$\mathbf{R}_{l_i,k}$	Landmarks observational noise covariance.

4.2 Towards playing soccer with much less use of localization

Human soccer players can effectively perform most of their tasks with a very poor estimation of their global pose in the field. They make extensive use of local information to: go to the ball, shoot to the goal, pass, keep close to the goal (in the case of the goalie), keep the ball inside of the field, mark opponents, etc. Even strategically positioning, which could be argued to be a localization-dependent task is performed with extensive use of local information; players do not only tend to be close to one static part of the field, they also (and may be more important) tend to maintain certain positions relative to their teammates, opponents and the ball. We believe our work is a step towards that direction, because it allows a robot to correct its odometry, and thus its relative estimates of non-seen objects, with local information.

5 Results

Preliminary results that illustrate the operation of the system are presented. Figure 3 shows a sequence of egocentric local maps, relative to the robot (coordinate system shown in figure 2), in selected moments of a real movement's sequence (data are collected from the robot and displayed in a visualization software). The sequence corresponds to the following situation: the robot walks from the yellow goal area to the center of the field, while panning its camera. In colors are shown the blue goal and the beacons (with exaggerated radiuses) estimations carried out by HL-Tracking. In the center of each map, the OEE particles appear, where lighter ones corresponds to those having a higher score. In order to make the functioning of the OEE visible, the particles' positions, relative to the center of the egocentric map, are zoomed ~4x with respect to the HL-Tracking estimation. In the tested sequence, the odometry was specially poorly calibrated, with a high bias (the accumulated odometry was of ~600cm, while the actual movement of the robot was of ~240cm). However OEE combined with HL-Tracking was able to keep tracking of the observed objects and correct the non-seen ones (yellow landmarks after they are left behind).

We have performed several experiments as the one already illustrated. In these experiments we have seen that when the robot perceives the fixed landmarks (the ones defining the map) regularly, the accuracy of the proposed robot's localization approach is similar that the one obtained when using a standard EKF, and no OEE or HL-Tracking stages (variations are less than 1% in accuracy). However, when the robot executes attention demanding tasks as approaching the ball without active vision behaviors for looking for the fixed landmarks, or turning with the ball while preparing a goal-kick, the accuracy of the robot's odometry estimation is 14% better than in the case when the OEE and HL-Tracking stages are not used, while the robot's localization is 6% more accurate.

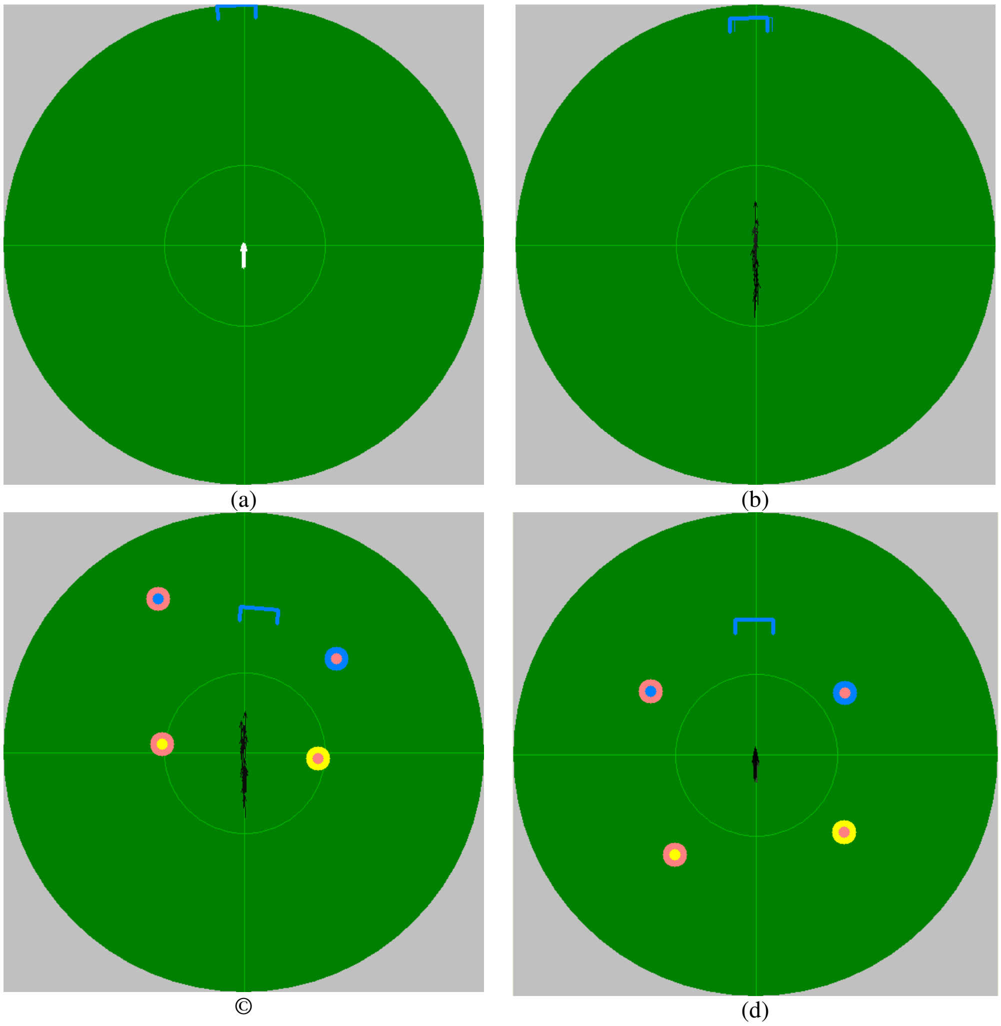

Fig. 3. Egocentric local maps in selected moments of a movement sequence: the robot walks from the yellow goal area to the center of the field, while panning its camera. a) The robot starts with a perception of the blue goal, all particles are together, with a high score. b) As odometry arrives, the particles start scattering, which allows the odometry correction. c) The robot is walking, in an intermediate point, d) The robot arrives to the center of the field.

6 Conclusions and Future Work

In this article, an improvement over the classical robot's localization approach was proposed, in which, in addition to the robot's pose, the robot's odometric error and the landmarks' poses are estimated. Based on this new approach, we developed an improved self-localization system for AIBO robots playing in a RoboCup soccer environment. In this system odometric error estimation is implemented using Particle Filters, while robot's and landmarks' poses are estimated using Extended Kalman Filters. Preliminary results show that, when the robot executes attention-demanding tasks, the accuracy of the robot's odometry estimation is 14% better than in the case when the new estimation modules are not used, while the robot's localization is 6% more accurate.

Currently we are carrying out a better characterization of the proposed system. In addition, we are developing an extension to the presented system, which consists in using the robot's odometric error for modeling and correcting the permanent odometric error using an on-line trained neural network.

References

1. Ruiz-del-Solar, J., et al., "UChile Kiltros 2007 Team Description Paper", *RoboCup 2007 Symposium*, July 9 – 10, Atlanta, USA, 2007 (CD Proceedings).
2. Guerrero, P., Ruiz-del-Solar, J., and Palma-Amestoy, R., "Spatiotemporal Context in Robot Vision: Detection of Static Objects in the RoboCup Four Legged League", *Proc. 1st Int. Workshop on Robot Vision* (*in 2nd Int. Conf. on Computer Vision Theory and Appl. – VISAPP 2007*), pp. 136 – 148, March 8 – 11, Barcelona, Spain, 2007.
3. Doucet, A., Andrieu, C., and Godsill, S., "On Sequential Monte Carlo Sampling Methods for Bayesian Filtering", *Statistics and Computing*, Vol. 10, no. 3, pp. 197-208, 2000.
4. Welch, G., and Bishop, G., "An introduction to the Kalman Filter," *Tech. Report TR 95-041* (Update May 23, 2003), Department of Computer Science, Univ. of North Carolina at Chapel Hill, 2003.
5. Isard, M., and Blake, A., "Condensation – Conditional Density Propagation for Visual Tracking", *Int. Journal of Computer Vision*, 29, 1, pp. 5—28, 1998.
6. Kalman, R.E., " A New Approach to Linear Filtering and Prediction Problems", *Trans. of the ASME - Journal of Basic Engineering*, Vol. 82: pp. 35-45, 1960.
7. Siegwart, R., and Nourbakhsh, I., *Introduction to Autonomous Mobile Robots*, MIT Press, 2004.
8. Polani, D., Browning, B., Bonarini, A., Yoshida, K. (Eds.), *Proc. of the RoboCup 2003 Symposium*, July 9 - 11, Padova, Italy, 2003 (CD Proceedings).
9. Nardi, D., Riedmiller, M., Sammut, C., Santos Victor J. (Eds.), *Proc. of the RoboCup 2004 Symposium*, July 4 – 5, Lisbon, Portugal, 2004 (CD Proceedings).
10. Montiel, J. and Davison, A., "A Visual Compass based on SLAM", *Proc. of the ICRA 2006*, May 15-19, Orlando, USA, pp. 1917- 1922, 2006.
11. Wang, Ch., Thorpe, C., and Thrun, S., "Online Simultaneous Localization and Mapping with Detection and Tracking of Moving Objects: Theory and Results from a Ground Vehicle in Crowded Urban Areas", *Proc. of the ICRA 2003*, May 12-17, Taipei, Taiwan, pp. 842-849, 2003.
12. Cheng, Y., Maimone, M.W., Matthies, L., "Visual Odometry on the Mars Exploration Rovers", *IEEE Robotics and Automation Magazine*, Vol. 13, No. 2, pp. 54- 62, 2006.
13. Konolige, K., and Agrawal, M., "Frame-frame Matching for Realtime Consistent Visual Mappin", *Proc. 1st Int. Workshop on Robot Vision* (*in 2nd Int. Conf. on Computer Vision Theory and Appl. – VISAPP 2007*), pp. 13 – 26, March 8 – 11, Barcelona, Spain, 2007.