HUMANOID ROBOT SELF-LOCATION IN SPL LEAGUE

Eduardo Munera, Manuel Muñoz, José Simó, Francisco Blanes
edmusnc @posgrado.upv.es
{mmunoz,jsimo,pblanes}@ai2.upv.es
Instituto de Automática e Informática Industrial AI2.
Universidad Politécnica de Valencia

Abstract

Mobile robot self-location is a classical problem in robotics since it was stated by Feng, Borenstein, and Everett [3] in the "Where am I" report, where methods for wheel robot location are presented using exteroceptive and proprioceptive sensors. But in humanoid robots proprioceptive sensors offer very limited information about real world position due to complex modeling physical interaction of feet plants and ground. In this case the use of exteroceptive sensors become essential for a correct location. In the case of the humanoid robots used in the Robocup Standard League (Nao robot from Aldebaran Robotics) the cameras located in the head are sensors where relay most of the information for the localization problem after a complex image processing and fusion. This ability it's more important in SPL since 2012 because changes in playing rules. These changes introduce a new playing field coloring with a symmetric environment making impossible to differentiate the opponent goal using colors like it was possible till now. From this perspective it's compulsory to avoid self-goal scoring to construct a global map of the environment using vision sensors, locomotion information in distributed and cooperating way between robots. This paper shows the results of the SPL Hidalgos Team in the development of the global map module to solve the self-location problem used in German Open 2012 games.

Keywords: Humanoid robots, self-localization, perception system, particle filter, Robocup SPL.

1 INTRODUCTION

During last dates, self-location has become important, as mobile robotics has been growing. Different kind of robots must be located to achieve satisfactorily the goals of its task. In the past, most of the applications were solved using own odometry, but this work become difficult when wheels skid due to the high velocities of work, or even more in legged robots. Thus, lots of methods have been developed to fusion kinematics with sensory information (inertial, visual).

The RoboCup Standard Platform League, is a robotic competition that faces in a soccer match, two teams of four Nao humanoid robots manufactured by the French company Aldebaran. Focusing our attention on this League, the location has become very important as other basic tasks have been achieved. Currently, being well located becomes a necessity, when you try to perform fluid movements through the field and plays in combinative way where more than one robot collaborate in achieving the task, scoring goals. This work has increased its complexity with the last changes in the regulation of this league, which remove the different coloring between own and opponent goals. In this way, no leave any fixed element for breaking the symmetry of the field.

This paper is organized as follow; first of all in section 2 have been made a compilation of related work, which provide an overview of the state of the art. In section 3 is introduced the perception system, it provides information about the surrounding environment of the robot. Next is proposed a new implementation the system based on other methods as is described in the section in section 4. Finally the results obtained from this work are detailed in the section 5 while the section 6 make as a briefly conclusion a review of all the described work and the advantages of it.

2 RELATED WORK

Implementation of reliable localization system is a well known topic in the RoboCup environment. That way several number of techniques have been developed and modified for a best improvement later.

Each technic have its own pros and cons as haven't been found yet a solution that disparages the others. Keeping this in mind there are a considerable variety of methods that have been used by the RoboCup's teams in the SPL has been show in [11].

One of the most used methods is the Unscented Kalman Filter (UKF) based localization system. As other types of Kalman Filters (KF) this technique offers a low computational costs and is represented as a normal distribution, which is parameterized as the Gaussian function. UKF have been proved to perform a better approximation in the prediction than others KF modifications by using a deterministic sampling technique to select minimal set of samples from the observations as shown in [9].

Another family of techniques that have been employed are the Particle Filter (PF) and its variants. The PF method represent an approximation based on a finite number of random samples in the workspace, which are characterized as particles. Each one of these particles has an assigned weight consequently to his probability of match the observation. Then a resampling process is applied to less weighted ones. Most of the PF implementations are based on the Montecarlo Particle Filter (MCL), which can be reviewed in [4]. The MCL method is computationally worst than the UKF but provide some advantages like more accuracy in certain cases and can deal with problems like kidnaped robot situations which are defined as an unexpected change of the robot position held by a referee or any situation caused by abrupt discontinuity in the robot translation.

There are multiple implementations and variations of the original MCL filter. Most of they are centered on the path to arrive to a sensor reset which will be explained in more detail and mainly allows the robot to find his real location once it has been lost, suffered a kidnapping robot situation or a fall of the robot itself. Some new features were added to the filter such as [7] in which different number of particles and numbers of random ones are taken into account.

Moreover an alternative oriented to the localization with ambiguous landmarks can be found in [2] where the dynamic progress of the filter depends on many parameters. Therefore optimizing these values is a critical step into the design of the filter as a result some research focused on this subject can be found. In most cases this values will be obtained by empirical tests but sometimes are used more advanced techniques like the procedure shown in [1] which apply the Particle Swarm Optimization.

3 PERCEPTION SYSTEM

Every kind of localization system must dispose of sensorial information about the robot surrounding to be able to determine the current position. Due of this the filter can set the probability of each position by comparing the recognized information with the information that are supposed to recognize at the estimated position. That information of the environment in most cases are relative to the field features that have been found which are characterized by a distinctive color, shape and position. That way on this section the vision sensor is describe showing its main features and its specific performance. The issues related to the field features recognition and the sensor adjust for reliable distance estimation have been detailed by introducing in each case the related problems and the solutions that have been found.

A new software tool have been developed to perform a calibration procedure to find both extrinsic and intrinsic parameters in an autonomous way independently of the robot is being used that way this process can be applied to each version of the humanoids robots used in the SPL.

3.1 VISION SENSOR

On Nao robot we have two cameras disposed on the head as can be observed on the Figure 1. These cameras are CMOS, which provides VGA images at a maximum of 30 fps. Both cameras have the same Field of View (FOV), which may vary depending on the used version just as the offset between the centers of the two cameras. The head have been assembled to the body trough a neck formed by two servomotors, which are responsible of the pitch and yaw movements. That movements allow the robot to perform a visual scan to recognize objects in a big range defined as the frontal side of a sphere defined by the configuration of the cameras and servos. It also is able to carry out visual servoing, which consists on track objects like the ball in order to keep it in his filed vision.

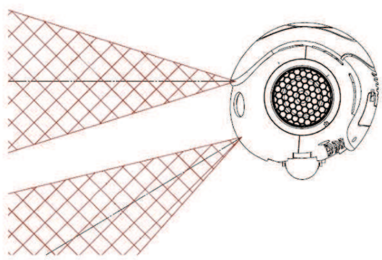


Figure 1: Vision sensor.

3.2 RECOGNITION SYSTEM.

The perception system that is describe here could recognize several landmarks by making a discrimination by color and shape. This has been designed to be configured through an external application that have been developed called H-Manager. This application allows the user to make an accurate calibration of the colors that must be recognize. The H-Manager also offers the possibility of specify or modify the profile of each landmark supposed to perceive.

Color calibration generates a LookUp Table (LUT), which are used in the segmentation process. This LUT is used in order to make image segmentation by color. Once we have the segmented picture image Blobs are formed by joining same color regions. Considering the generated blobs this system can recognize the related landmarks of the valid ones as has been shown on the Figure 2 and discards invalid ones. As has introduced before this system is able to offer enough information to play reactively so that it could deal with reactive playing control just doing a color discrimination.

3.3 SENSOR MODELING AND ADJUST.

There are a considerable amount of researches focused on camera based distance estimation, an example can be found on [6]. The very first step for being able to measure distances with the vision system consists of having extrinsic and intrinsic camera parameters properly calibrated to reduce errors in distance estimation.

Once camera sensor has been modeled and adjusted the two procedures shown below can be combined in order to have a right measure system, that way is a trivial step making a distance estimation function which allow to obtain the polar position of a particular environment feature by a simple triangulation as show Figure 3 taking into account the head servos positions and the height of the camera, which rely on the kinematic model of the robot as will be exposed on the extrinsic parameters section, and also the position of the concerning feature in the image.

Is important to point that the Nao's software runs on a middleware provided by Aldebaran Robotics that offers among many other things information about the position and angles of each servo and relatives positions like the height of the camera in the robot local system.

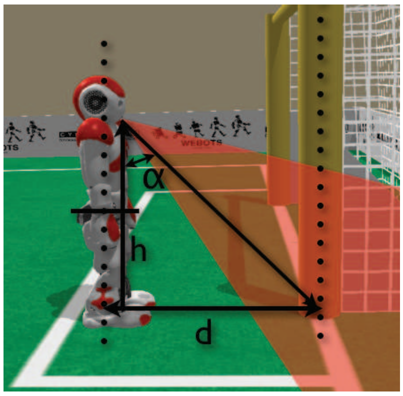
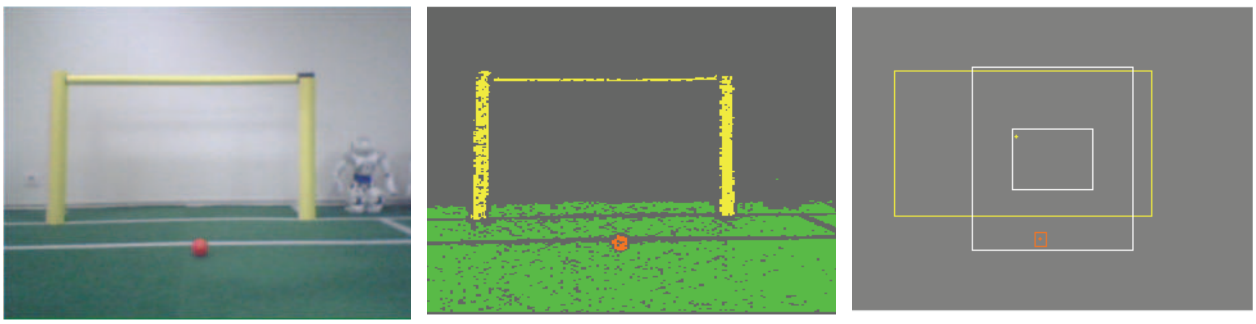


Figure 3: Rho estimation.

3.3.1 Extrinsic parameters adjust.

The extrinsic parameters of the camera describe the coordinate transformation between the camera and the given point of reference in the image. As we can see in [6] these parameters are strictly dependent on camera position. In this way the distance measured is directly related with head position, which relays on the configuration of the servo-motors in every control period. Assuming that a study centered on the characteristic profile of each servo involved on the camera movement must be done. The profile of the



a) b) c)

Figure 2: a) Original picture. b) Segmented. c) Blobs.

servos position has been introduced as the most responsible to obtain the equivalence between an image point and a 3D world one. In this way studying the behavior of the motors and his relation to distance of the real feature have been revealed as a worth technique for a fine adjust.

For this analysis is needed to characterize the relation as the result of a sine function of the head angles. In the case pitch angle values close to π/2 the resolution obtained will provide very low deviations for large distances. Due to this problem the head pitch angle is considered the main subject to adjust. This adjust will rely in that naoqi's kinematic system can provide a mostly reliable measure of the camera height and the value of head yaw angle can be modeled with a fixed offset.

Doing an empirical test the profile of the camera pitch movement can be obtained for every robot associating the central point of the image to the correspondent distance in the real world. In the Figure 4 we can realize that a polynomial function can be found as the most simple function to adjust the shape of the empirical representation giving us the capability of make a forthright conversion from angle to distance. This procedure can be applied for each camera taking into account that both of them depend from the same motor and consequently present the same profile with the offset calculated.

3.3.2 Intrinsic parameters adjust.

Intrinsic parameters are required to compensate the distortion effect produced by the camera lens. Although there are several relevant method like Hekkila [5] or Tsai [10] the selected method will be based on the Zhang [12] procedure as a result of the comparison that have been made in [8].

By extrapolating the previous conclusions obtained in 3.2.1 we can directly assume that there are a relation between the number of pixel in the image and a certain angle of separation between them and the real position of the camera in the 3D world. This premise allows calculating the equivalent camera angles for a non-centered feature found in the image as if it was. The right equivalence to make the pixels-angle conversion is conditioned by the angles of aperture of the camera lens whose have to be take into account in both his vertical and horizontal axis and calculate its relation with the resolution size of the picture. In this case unlike what have been done in the extrinsic case the top and the bottom camera are processed each one by its part because the correction of the intrinsic values are applied separately in the bottom and the top camera.

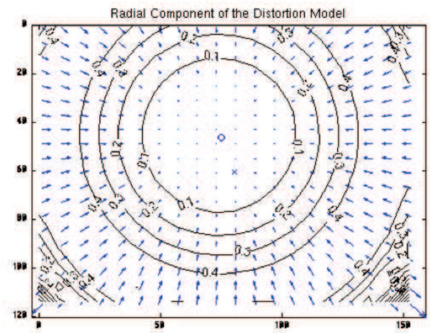
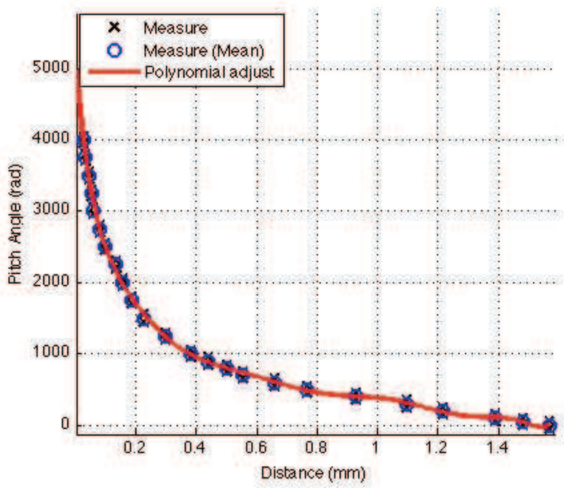
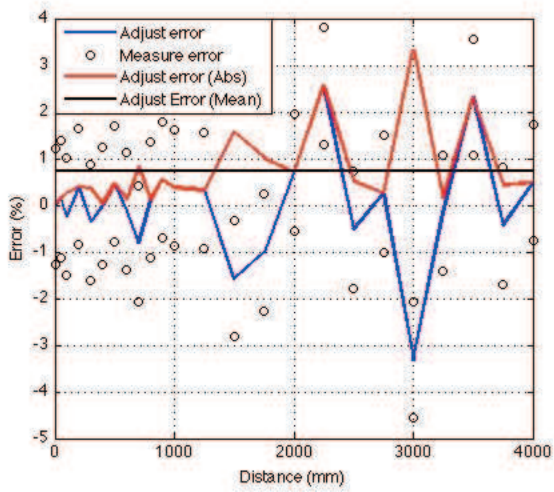


Figure 5: Radial distortion of the Nao camera lens.

Although this relation can be made this is not a proportional or lineal relation on the image distortion. It can be defined as barrel or as pincushion due to the distortion shape as it is defined in [12]. A exhaustive analysis of the Nao camera distortion is showed in Figure 5. The distortion must be deal on each axis, as lenses haven't to be symmetrical on each one. In this case was being applied the



a)



b)

Figure 4: a) Empirical measures of camera pitch (blue) and polynomial adjust (red). b) Adjust error.

"estimating radial distortion by alternation" method as is described on [12] on the "dealing with distortion". This method involves three parameters for each axis as show on (1).

$$\begin{bmatrix}(u-u_0)(x^2+y^2) & (u-u_0)(x^2+y^2)^2 & (u-u_0)(x^2+y^2)^4 \\ (v-v_0)(x^2+y^2) & (v-v_0)(x^2+y^2)^2 & (v-v_0)(x^2+y^2)^4\end{bmatrix} = \begin{bmatrix}\breve{u}-u \\ \breve{v}-v\end{bmatrix} \quad (1)$$

New capability for dealing with the distortion has been implemented in the system according to the radius from the image center to the pixel on interest.

3.3.3 Auto-adjust system.

Obtaining the extrinsic and extrinsic parameters has been introduced as the clue of the success of the distance estimation system. Despite of this must be considered that each robot is susceptible to have some differences in the value of its configuration parameters compared to the rest. These values also can be modified over time due to mechanical damages because of falls of the robot. That's the main reason that lead us to design an auto-adjust system for make each robot capable to obtain its own parameters by the executing of an autonomous procedure. Focusing on this idea we developed an OpenCV based system which main function is to obtain this parameters and store them on the robot memory on a XML file in the robot memory allowing to load this configuration each time the robot is restarted. This adjust in both (extrinsic and intrinsic) types of parameters consists on the placement of the robot in a well-known position and once there make it recognize some points on the field through its OpenCV application.

The auto-adjust system is designed to be a new functionality offered by the previously HManager tool mentioned. The developed algorithms are not running in the robot hardware, instead of this the application apply for a picture and later the image process was made on the computer trough the HManager communication, after that calculates the new position for each head servo, and send the information to the robot again. Thanks to this application all the adjust process can be supervised on the computer working on a distributed and supervised way in a minimum time.

First of all the extrinsic parameters will be obtained based of the idea that each motor is adjusted to the profile close to the one showed in the Figure 4 a). With this in mind the adjust parameters can be obtained by determining the polynomial function to the correspondent profile of the implied motor. For this task the robot finds a set of points in the field that must match with the horizontal field lines in order to dispose the needed measures to fit the required adjust function. Using this strategy the lines in the game field are used as references for the calibration process. In order to find the position which offers the best calibration through the lines that can be spotted have been implemented an optimization method which provides the required location by evaluating all the potential positions.

This calibration procedure was applied on the head bottom camera and taking into account that the movement of the upper camera depends on exactly the same motor the previously obtained adjust function can be applied to its measures just adding the offset between the two cameras. For a better performance this offset is obtained by finding the same point with both upper and lower camera and computing the difference into the reached positions.

For the radial distortions adjust parameters the first step on the used method consists on estimate the distance to an image-centered feature which is supposed not to be affected with any kind of distortion. Once this pint is reached the camera must be moved to find the same feature but now it have to be on a non-centered position. This operation is repeated several times in some image positions that must be at different radius from the image center and represent all the range of values. By doing this process ensures to bring information of different radius of distortion of the camera allowing applying the Zhang expression (1). This time the vision system have to be capable to find a certain point on this image, beside of this it was defined to find the corner of a L shape as can be observed in the Figure 6.

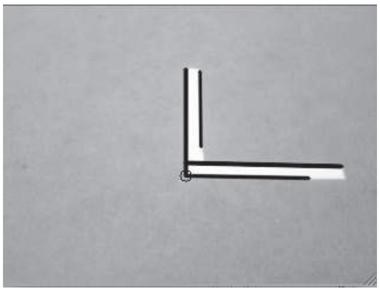


Figure 6: L shape recognition.

4 LOCALIZATION SYSTEM

At this point all the necessary correction and analytic techniques to get good distance estimation to field features have been detailed on the previous section. Therefore in this section will describe all the elements that define the filter used for robot self-location, which has been anticipated as a modified version of the Particle Filter.

4.1 LANDMARKS.

The particle filter bases its way of work through the information provided by the perception system as has been described before. In spite of this there are some information that is not relevant and haven't to be processed. The vital information to make filter determines the real position of the robot is: the goals, the borders lines of the field and the information shared by others robots of the same team. Other field lines have been discarded because its computational cost was not compensated with a relevant improvement of the estimation. But future developments on new Nao versions, which are equipped with powerful CPU, open new possibilities for improvement.

The goal position brings in most cases the necessary information to determine the position of the robot in the field. The measured position of the goal have been defined as a relative distance between the robot current position and the goal and the inclination angle between the robot orientation and the imaginary axis passing through the crossbar in global system. By considering this information there is only a unique position that will be coherent with this measures resolving all the uncertain about the robot position.

Border white lines give information that helps to locate the robot position in a point of the parallel line at a measured distance with a orientation based on the line inclination. That only provides a roughly approximation of the real position but offers a good information in that cases that the robot can't see the goal preventing the particles to get scattered all around the field.

Although all of this offers a right localization this isn't enough to make the robot solve the problems caused because of the game field symmetry, namely with no previous information there is no way to discern which middle field is playing (if its own or opponent) no matter how good is the sensed information. By introducing shared information (using communication between robots) robots with a reliable position can provide his own information helping the receiver robot to quickly converge to a unique position by discarding the symmetric one by incoherence.

4.2 MODIFIED PARTICLE FILTER.

In this section we proceed to sketch the algorithm that have been developed for robot self-localization. As have been introduced this is based on a modified version of the Particle Filter introduced previously taking into account certain aspects of the augmented filters that have been developed by other RoboCup teams.

Since now landmarks have been declared as the core of the localization system. In this way this implementation of the filter leads to a solution based on the coherence between all the spotted features. As can be noticed the in the Algorithm 1 each particle is updated using the odometry information and a random noise, then calculate the error associated for the measured distance to each feature and distance from the particle position. Assuming that the less error value for all particles on each featured was stored. Through these values a new reference particle can be obtained by finding the coherences between them.

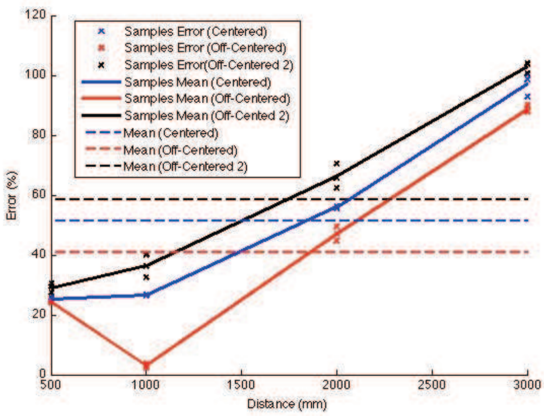
Algorithm 1: Modified Particle Filter

```
for i=1 to N do
    p(i) <- odometry_update(p(i), odom)
    for j=1 to Num_Features do
        error <- error2Feature(p(i), feat(j))
        if error < best_e (j)
            best_e (j)<- error
        end if
    end for
end for
refP <- coherence_features(feat, best_e)
w_threshold <- sensor_reset_threshold(best_e)
for i=1 to N do
    w(i)<- scale_weights( p(i), refP)
    if w(i) < w_threshold
        p(i)<-sensor_reset
    else
        p(i)<- resample(p(i), w(i), refP)
    end if
end for
```

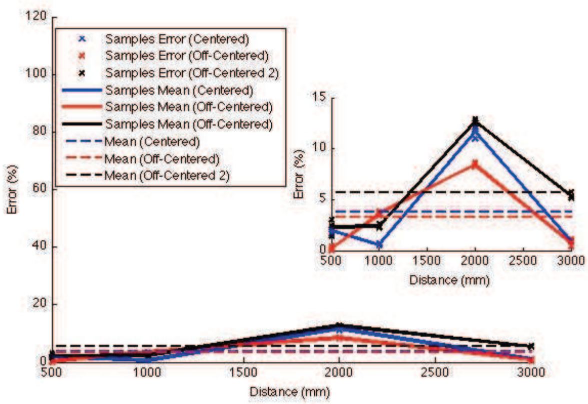
The main difference between the presented implementation of the particle filter and others is that the weights that are being assigned are not based on the probability of each particle to be in the supposed location but based on how close is each one to the reference particle that has been calculated. That way the sensor reset is dependent on how big is the best error. In case of a low error value most of the particles values must reach gradually the real position by updating their position to locations near the reference position, which grants the best error situation. In case of the reference particle is associated with a high error and the value of the threshold for avoid sensor resetting will be increased forcing to generate new random particles on all the field model allowing a new one to offer the best error situation and making old ones converge to this new position.

5 RESULTS

In this section it will be evaluated the sensor model and adjust described in section 3. As have been observed one of our goals to achieve with this adjust is the capability of measure distances between the robot position and a spotted feature. That way has been prepared a simple experiment in which the distance to a same feature have been measured in some different real distances between 500mm and 3000m and three different positions in an image, one centered and two off-centered, the first with the feature located on the lower half of the image and the second on the upper one. This experiment must be carried out two times, the first without the designed adjust, procedure in which the distance is obtained using a simple triangulation, and the second time with the adjust method proposed. Checking the Figure 7 can compare the results.



a)

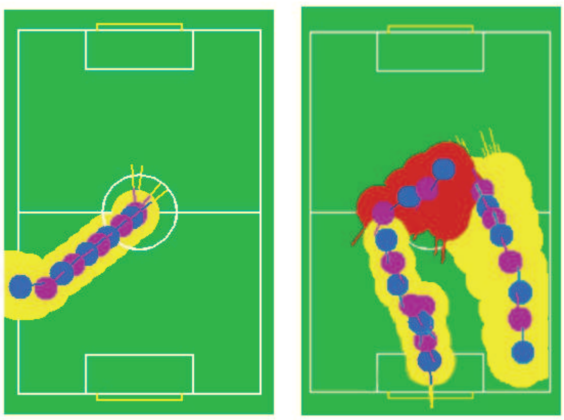


b)

Figure 7: Tests results: a) without adjust. b) with adjust.

Must be noticed that in the first test, showed on the Figure 7 a), the induced error is enough for provide a not precise estimation when dealing with short distances but still is an indicative measure. Despite of this for distances above 1000mm the estimation does not provide any reliable information due of the estimation error. Taking this into account the mean error obtained between the range of distances used in this tests can be up to the 60% but for some measures have been raised errors above the 100%. Obviously this estimation is characterized as an unusable method for the distance estimation is wanted to perform a precise localization system for world modeling.

Unlike the unadjusted method can be observed in the Figure 7 b) how with the designed method the obtained error is not dependent of the distance that is being measured. The maximum mean error that has been obtained is around the 5% of the measure and the error of the sampled measures is at most around the 10%. As a conclusion this method offers the needed precision for make the particle filter able to percept a similar model of the real environment. On the other hand is being evaluated the performance and the advantages that provides the introduced method in comparison with the odometry.



a) b)

Figure 8: Localization result: a) Positioning test. b) Free play tests.

In order to check the localization system has been carried out some tests in which have been used the proposed implementation of the particle filter. In the first one as can be seen in the Figure 8 a) the robot must be positioned in the center of the filed oriented to the opponent goal. In the second test the robot performs a free play situation in which is forced to describe a trajectory similar to the one observed in Figure 8 b). In both tests can be appreciated how the particles are being spread around the robot and how the estimated position coincides with a certain tolerance with the real one. Relying on the results can be proved how the developed filter offers a better position and orientation estimation than the test in which only have been used the odometry.

6 CONCLUSIONS

As conclusion we want to highlight that we have introduced a complete procedure to implement a reliable self-localization method ranging from the perception system to the localization system itself that it has been applied for the robots used the SPL League, but could be generalized for other humanoid robots.

First it has been developed a sensor modeling method and it subsequent adjust to improve the measuring system and the perception of the environment. As has been shown in the results this adjust can evaluate the error produced in each measures because of this previous modeling process allowing to correct it and consequently offering a good distance estimation.

On the other hand it also have been introduced a new implementation of the particle filter whose operation lies on finding the best particle in every cycle by evaluating the coherence to each feature and checking the coherence between them. This best particle determines the sensor resetting process in case of a bad location in the game field.

Finally that system has been evaluated to offer a better localization than other systems like the odometry allowing the robot to know his estimated position at any time and maintain the correct orientation as main reference to determinate at what side of the field is the robot looking and consequently realize if must kick to that direction instead of the opposite one. Therefore this system provides an essential element in order to improve the game and provide a real challenge for the opposing team.

Acknowledgements

This work was supported from the Spanish MICINN project COBAMI DPI2011-28507-C02-01/02 founds.

References

[1] A. Burchardt, T. Laue, T. Röffer, (2011) “Optimizing Particle Filter Parameters for Self-localization”, *LNAI 6556, pp. 145-156, Germany*.

[2] B. Coltin, M. Veloso, (2011) “Particle Filter-based State Estimation in a Competitive and Uncertain Environment”, *Universtät Bremen, Germany*.

[3] Feng, L., Borenstein, J., and Everett, B., 1994, *"Where am I? Sensors and Methods for Autonomous Mobile Robot Localization." Technical Report, The University of Michigan UM-MEAM-94-21, December 1994*.

[4] D. Fox, W. Burdgard, F. Deallert, S. Thrun, (1999) “Monte Carlo Localization: Efficient Position Estimation for Mobile Robots”, *Proc. of the National Conference on Artificial Intelligence, United States*.spo

[5] J. Heikkilä, O. Silven, (1997) *“A four-step camera calibration procedure with implicit image correction”, CVPR97*

[6] T. Laue, T. Jeffry de Haas, A. Burchardt, C. Graf, T. Röfer, A. Härtl, A. Rieskamp., (2009) “Efficient and Reliable Sensor Models for Humanoid Soccer Robot Self-Localization”, *Proceedings of the 4th Workshop on Humanoid Soccer Robots, Germany*.

[7] T. Laue, T. Röfer, (2007) “Particle Filter-based State Estimation in a Competitive and Uncertain Environment”, *Universtät Bremen, Germany*.

[8] F. Remondino, C.Fraser., (2006) “Digital camera calibration methods: considerations and comparisons”, *IAPRS Volume XXXVI, Part 5, Dresden*.

[9] S. Thrun, W. Burgard, D. Fox, (2005) “Probabilistic Robotics”, *MIT Press, pp. 69-70, Cambridge, Massachusetts*.

[10] R.Y. Tsai, (1987) *“A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses”, IEEE Int. Journal Robotics and Automation, Vol. 3(4), pp. 323-344*

[11] H. van der Molen, (2011) “Self-Localizarion in the RoboCup Soccer Standard Platform League wuth the use of a Dynamic Tree”, *University of Amsterdam*.

[12] Z. Zhang., (2000) “A flexible new technique for camera calibration”, *IEEE Transactions on Pattern Analysis and Machine Intelligence, vol 22, pp 1330-1334*.