A Review on Improvement techniques for Self-Localization in RoboCup Soccer

Nitish A. Gupta
Department of Electrical Engineering and Computer Science
University of Central Florida
Orlando, Florida 32816, USA
nitish.gupta@knights.ucf.edu

Abstract

Self-Localization of the robots has been one of the challenging and widely researched areas in RoboCup Standard Premier League (SPL). The main problem occurs when the robot gets kidnapped by losing its confidence values which might occur due to falling, collision, symmetry of the field or obstruction caused by other robots. Moreover from the past few years of SPL the field setups have evolved in a way that it becomes less informative than previous year. This results in decrease in artificially perceivable objects and landmarks and increase in ambiguity. For the underlying localization algorithms to perform well, these ambiguities have to be resolved before apply the algorithm. Several different approaches have been proposed in order to make self-localization independent of artificial landmarks and decrease the ambiguity. In this paper we will discuss some of these successful techniques.

1. Introduction

The localization algorithms like Monte-Carlo, Kalman filter or Particle filter localization has several variations and are used widely in estimating position and orientation of a robot on the field. The measurement update step compares already learned features from a map to that with the features (static or non-static) perceived from environment. The main question here is which feature should be perceived from the environment so that the robot could make an unambiguous decision and accurately localize itself.

One of the immediate observations of the field would suggest us to use field features like lines, intersections and circles [1] is discussed in section 2.1. Although this method has proved to work up to some extent, it largely depends on static landmarks. Thus when the field lines are obstructed by another robot or any other object, this approach would not perform as expected. Another approach would be to predict feature observation given a robot motion [2]. This approach uses the relations or bearings to horizontal landmarks given the knowledge about the robots movement between the observations of the features. If a relation between two observed landmark features could be found, given the knowledge about robots motion, then we could predict the landmark as a whole and thus self-localize. This idea is presented in [2] also discussed in section 2.2, where the authors make use of odometry and horizontal bearings to landmarks to self-localize the robots in SPL.

In [3], an approach were a probabilistic model is developed to identify non-unique landmarks using importance sampling technique followed by hypothesis pruning is presented and is detailed in section 2.3. Apart from perceiving landmark or visual features, some approach to localization problem is based on perceiving Color features. For instance, in [4] (discussed in section 2.4), Markus et al. uses a separate Room-Awareness Module to perceive color histograms which helps Behavior controller to make corrections in position or orientation. These histograms are then compared with a background histogram model to evaluate and rectify the pose confidence of the robot. A different and interesting approach was taken by Ahmet et al. [5], also explained in section 2.5, wherein they used perception of another robot to help a lost robot self-localize itself. This technique is called collaborative localization and is suitable in multi-robot environment like in SPL.

2. Related Work

2.1. Field Features

Many variations of self-localization technique using field-lines feature detection have been used in the RoboCup SPL since its first implementation in Sony Four-Legged Robot League. Originally this approach used field-lines intersection (corners) detection instead of using field-lines detection which is a costly process in time and memory. The idea is to extract segments from images and group them together in a set of features in Local Perceptual space (LPS).

The input images are given in YUV format which are then mapped to a horizontal frame with filtered borders using Sobel filter. These mapped images are then converted to HSV space where the transitions can be extracted easily. These transitions are then labeled and grouped together to get straight line segments using Recursive Iterative End Point Fit Algorithm. Now the intersection of these segments is labeled as closed, open or net, depending on the type and angle of intersection. These labeled intersections are further grouped together into Type C, T-field or T-net based on the proximity of the labeled intersections to get the final set of features in LPS space. The distance and orientation of a robot from a particular intersection is calculated using projection of a pixel on the feature.

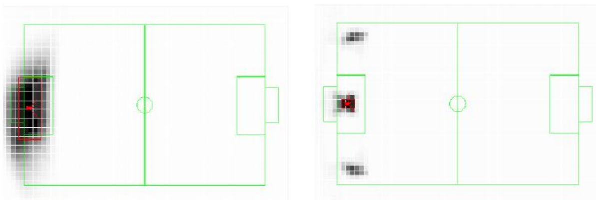


Figure 1: Resulting beliefs (a) Coloured landmarks and nets (b) Field lines and nets

This feature extraction technique combined with a localization filter like Fuzzy-Markov can be used as effective self-localization approach. In the experiments performed by the authors in [1], they placed the robot in front of a goal area with belief evenly distributed across the field. The results showed that colored landmarks based technique had a higher uncertainty across the goal post as compared to the field-lines based technique also shown in Figure 1. Also the absolute position error in field-lines and net based approach was as less as 100 mm to that compared with colored landmark based approach.

2.2. Landmark Bearings and Odometry

Bearing and odometry based self-localization techniques have been widely researched. Since this technique perceives angles to the landmarks instead of image features, the localization is not dependent on internal representation of position estimate which increases its robustness as compared to other techniques. To understand this technique in more detail, let us consider the well-known bearing and odometry based research in [2]. The method requires only two inputs, information about the bearing angle α and a motion vector which calculated using odometry data. These two vectors are calculated at different times in space and stored in a small buffer for further analysis. The goal here is to find a likelihood function that gives possible estimates of robot position (x, y). Later this likelihood function is optimized using maxima of a function to obtain the best position estimate.

Given the bearing angles to different landmarks and position of each landmark, an average angle is calculated between horizontal bearings. Now the sum of squared difference between the average angle and each bearing from landmark gives a likelihood function when the robot is not in motion. This technique as shown in Figure 2, is known as pose estimation using angular constraints. In order to take motion into picture, the motion vector from odometry data is incorporated in the above calculated likelihood (or similarity) function by using simple trigonometric relations. The similarity function can incorporate a number of observations from the past. When maximized using one of the optimization techniques like gradient descent, this function can provide a fair estimate of robots position.

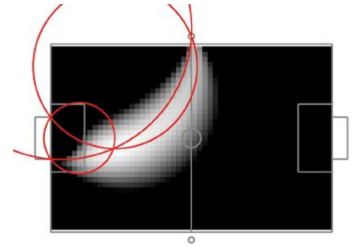


Figure 2: Pose estimation using angular constraints

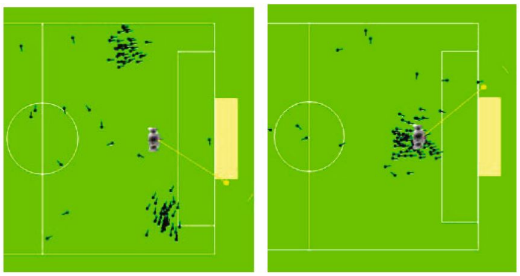
In an experiment, this template generation technique was used with a multi-hypothesis localization algorithm Monte-Carlo with 200 particles distributed initially. When the template generation was not used, random jumps of position estimates from the ground values were noticed. In contrast, it was closer and smoother with template generation approach. When large numbers of such sample templates were used, the position error was close to 3%.

2.3. Non-Unique Landmarks

This method works on the assumption that, if the robot has observed a particular feature than the next feature observation would be associated with the previously observed features. This approach is known as Multi-Hypothesis Tracking (MHT), where association of observed features forms the hypothesis which is tracked using Kalman Filter. In a localization algorithm, after each motion update we compare the observed input with the stored landmark in the map at that position, to update the belief.

In this MHT, instead of storing a single landmark of a particular object, we store a subset of landmarks associated with that object like left yellow bar, upper yellow bar and right yellow bar. Multiple numbers of such groups are associated with the map. Now while localizing, if the robot first observes a left-yellow and after a motion update it observes a right-yellow bar, then it would be able to localize itself if the number of observed features in consecutive motion fall into any of the subsets stored in the map. This is implemented by calculating the probability of partially observed features being from one of the subset groups in the map. If we model our hypothesis as Gaussian, after each observation, the number of Gaussians increases exponentially, giving a mixture of Gaussians after some observations. Thus resampling of bad hypothesis is essential step, also called as pruning. Thus the algorithm converges towards optimum hypothesis in a non-unique landmark environment.

This model was used in RoboCup SPL with promising results. The robot was able to perceive goal bars and calculate distance to the goal. In an experiment when the robot was placed in front of a left yellow bar, two hypothesis where generated for being at each goal post. But as soon as the right yellow bar was perceived the wrong hypothesis was pruned and weights of the correct hypothesis were increased. Similarly, the robot was also able to perceive corners based on their shape, T or L as shown in Figure 3. Initially the particles were concentrated on either side of the goal post, however as soon as the second corner was perceived, the



(a) During first observations (b) After convergence

Figure 3: Non-Unique Corner Detection

particles converged at the center.

2.4. Spontaneous Reorientation using Colored histograms

Spontaneous Reorientation is intuitive approach which is inspired by psychology findings of how humans perceive subjective geometric impressions of their background rather than object detection. For perception, this approach uses color histograms mapped on a virtual wall around the field in a two-row cylindrical shape. This perception is calculated using a separate Room Awareness Module (RAM), which works along with a Self-Localization and a behavior module for making correction in the position. The self-localization module allows RAM to update the confidence values through Behavior controller in one cycle.

The integration of RAM with self-localization module allows Behavior controller to issue commands like flip, purge or reset pose on receiving the pose confidence values from RAM. Due to symmetry of the field, the color images are also symmetric but their histogram breaks this symmetry. The color histogram is linked to the tiles of the surrounding virtual wall using 13 color-bins. This background model in RAM is trained online with corresponding histograms using a moving average update strategy. The new variance and mean is calculated for each new measurement. If the variance is too large, the histogram is not recorded. In RAM, the background evaluation is done using robot pose and perceived histograms. These perceived histograms are compared with the model histograms to assign weights to the particles in a particle filter algorithm. If a best match is found, new particles are added to avoid local minima. The visualization of the color histogram can be seen in Figure 4.

During the experiment, a real robot was placed on a field with incorrect initial pose. Initially it kept wondering until finally it learned the histogram model after which it predicted correct position. Because of the online learning, the robot fails to localize 15% of time but once the model has been developed, RAM indicates high confidence value. Similarly when the robot is placed with incorrect pose, the particle starts to grow in both possible location due to symmetry, but as soon as RAM identifies the goal view, the other hypothesis is purged immediately.

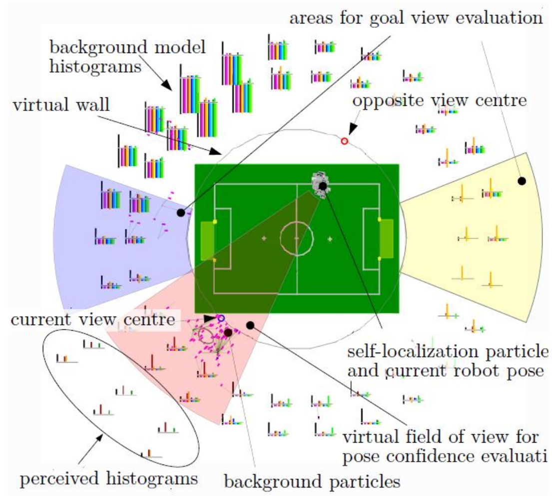


Figure 4: Background model

2.5. Collaborative approach

In RoboCup soccer, the robots on the field are connected to each other through a wireless network for communication. Network localization of stationary robots is common and widely research area in Multi robot environment. Taking this idea a step further, the authors in [5] present a Collaborative localization approach of mobile robots to resolve the ambiguity in self-localization. The goal here is to merge the perception of all the robots to get a common world perception. This is done without identification, such that a robot can only perceive a teammate but cannot distinctly identify it. Also the self-localization of each robot is noisy, thus the perceiver's position must be reliable.

This approach uses message communication to send information like its player-number, orientation, mean-variance, number of robots in its view and their relative distances between them from its perspective. Using the mean as the center of each robot's position, circles are used to represent each robot. The type A-circle is the robot's belief position; the B-circle represents its perception of a teammate's position and the O-circle is the correct position of each robot obtained after pose and distance correction. The radius of these circles is proportional to the reliability of its or its perceiver's belief which has a negative correlation with the variance of position belief. Each robot has its own A-circle as well as one or multiple perceived B-circles which intersect with each other. We obtain O-circles by merging the A and B circles using few rules. These O-circles have higher reliability then their parent circles. Map-merging is followed by Rotation correction, wherein the lost robot's orientation is rotated to obtain a correct orientation based on its perceiver's belief. Using a particle filter, weights of the particles in the O-circle is increased after the robots have obtained an O-circle to help their belief converge.

Using experiments results in [5], it was shown that MRL improves self-localization of robots in simulated as well as real environments. The improvement in position was as large as 50%, while improvement in orientation was not as signif-

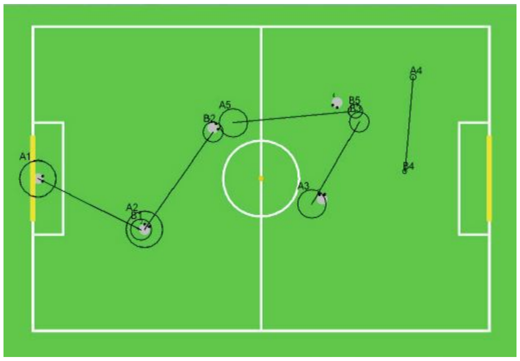


Figure 5: A, B-circles

icant. Also the performance in simulated environment was better than that in a real environment due to the noise.

Conclusion and Future Work

The approach to the solution in each of the techniques explained in this paper may be different however the goal is common, to remove ambiguity in self-localization. The field-lines based approach is considered to have low computational burden but it is prone to false positives due to line color dependency. The bearing and odometry based approach on the other hand, doesn't require internal representation of the position but faces large errors in odometry due to influence of opponents. The Non-unique landmark based approach is complex in nature; also initial training of the transition model could be cumbersome. The Color-Histogram based approach exploits the unique non-perceivable features of environment, like illumination to break the symmetry of environment. However, questions like - the total number of color histograms required to perceive features uniquely and optimal search of distinctive background remain unanswered. The Multi-robot based collaborative approach gave promising results in simulation, however, in the real environment the improvements were not significant due to noise.

These algorithms will not only have to overcome their own limitations but also the challenges that are yet to come as the competition keeps evolving towards the real soccer.

References

[1] Herrero-Perez, D., and H. Martinez-Barbera. 2006. *Robust and efficient field features detection for localization.*

[2] Jungel, Matthias, and Max Risler. 2007. *Self-localization using odometry and horizontal bearings to landmarks.*

[3] Ozkucur, N. Ergin, and H. Levent Akin. 2011. *Localization with non-unique landmark observations.*

[4] Bader, Markus, and Markus Vincze. 2013. *Spontaneous Reorientation for Self-Localization.*

[5] Erdem, Ahmet, and H. Levent Akin. 2014. *Multi-robot Localization by Observation Merging.*