729G50 ARTIFICIELL INTELLIGENS

RoboCup

Robots acting individually aswell as part of a team

Bertil Carlsson, berca955@student.liu.se, 860227-4071

2007-10-12

Linköpings Universitet HT-07
Linköping

Bertil Carlsson, berca955
860227-4071

Abstract

This article focuses on two things, a model of a Multi-Robot System. Here I explain the taxonomy of such a system and explain a bit on how this is implemented in robots playing football. Secondly I explain a relatively new self-localization algorithm which has shortened the computational time for robots prominently.

Bertil Carlsson, berca955
860227-4071

Context

Bertil Carlsson, berca955
860227-4071

Introduction

The goal of RoboCup is that by around 2050 a team consisting of fully autonomous humanoid robots should be able to win a match against the team which won the last FIFA World Cup. The rules of the game are the same as the official rules of FIFA. Therefore in 2002 the RoboCup Federation added a humanoid robot league to their annual football championship.

The RoboCup humanoid League completion rules state that the robots must have a human-like body plan. This means that they have to consist of a trunk, two legs, two arms and a head. And the only movement allowed is by the use of bipedal walking. Also the robots must be fully autonomous, no external power, computing power or remote control is allowed. Everything must be applied to the robot beforehand.

With these requirements come a lot of problems which have to be solved. One of them is getting the robot to be able to stand on one leg while kicking a ball. However first the robot should be able to walk, localize the ball, localize other players on the pitch and being able to know where on the field it is. In this article I first take a look at how a Multi-Robot System (MRS) could work and be implemented into football playing robots.

I then move on to take a look at a fairly new algorithm which have been developed for the purpose of making the robots more autonomous on the pitch. This acquired mainly by shortening the computational time, making it better than most self-locating algorithms used in RoboCup.

Multi-Robot systems (MRS)

The Multi-Robot System (MRS) is handled as a form of Multi Agent System focusing on reactivity and social deliberation. The MRS can be used for a number of things, e.g. foraging and box pushing, however here focus lies on football -playing robots.

L. Iocchi, D, Nardi and M. Salerno describes the taxonomy of the MRS in their article *Reactivity and Deliberation: A Survey on Multi-Robot Systems (2001)*. They suggest via a top down approach that there should be four levels individuated in the taxonomy, and they are *a Cooperation Level, a Knowledge Level, a Coordination Level* and an *Organization Level.* Communication and system composition can also be added to the taxonomy however they characterize the system independently.

The taxonomy

The cooperation level

At the Cooperation Level they distinguish between cooperative systems and not cooperative systems. They define cooperation as a situation in which several robots work together to perform a global task which cannot be solved by a single robot or whose execution can be improved by cooperating with other robots. Robotic agents operating in the same environment without any relation to each other which involves a common objective (*not cooperative* MRS) are therefore not taken into account when referring to MRS in the following. The first important characterization to be made is how much the robot knows about the presence of other robots in its team, which leads to the next level in the taxonomy, the Knowledge Level.

The knowledge level

L. Iocchi, D. Nardi and M. Salerno talks about different levels of awareness. Robots can perform a task in an environment without having any knowledge about other robots in the system, this is a very weak form of cooperation. For instance box-pushing robots can perform their task of pushing boxes without having to interact with another agent. This does however not work when two football playing robots are to pass the ball to one another. Here they use direct communication to know where the other robot is located in order to perform a successful pass. Therefore they introduce a third level in their taxonomy: *The Coordination Level*.

The coordination level

Coordination is explained as a cooperative state in which the action of every robot takes into account every action performed by the other robots in the team. This so that the whole ends up being a coherent and high performance operation. Robotic systems in which coordination between systems is required, but where the robots are not sharing the same goal are excluded in L. Iocchi, D. Nardi and M. Salerno article. They also make a difference between coordination that relies on a coordination protocol and a coordination which does not. Calling the latter weak coordination and the other for strong coordination. Strongly coordinated

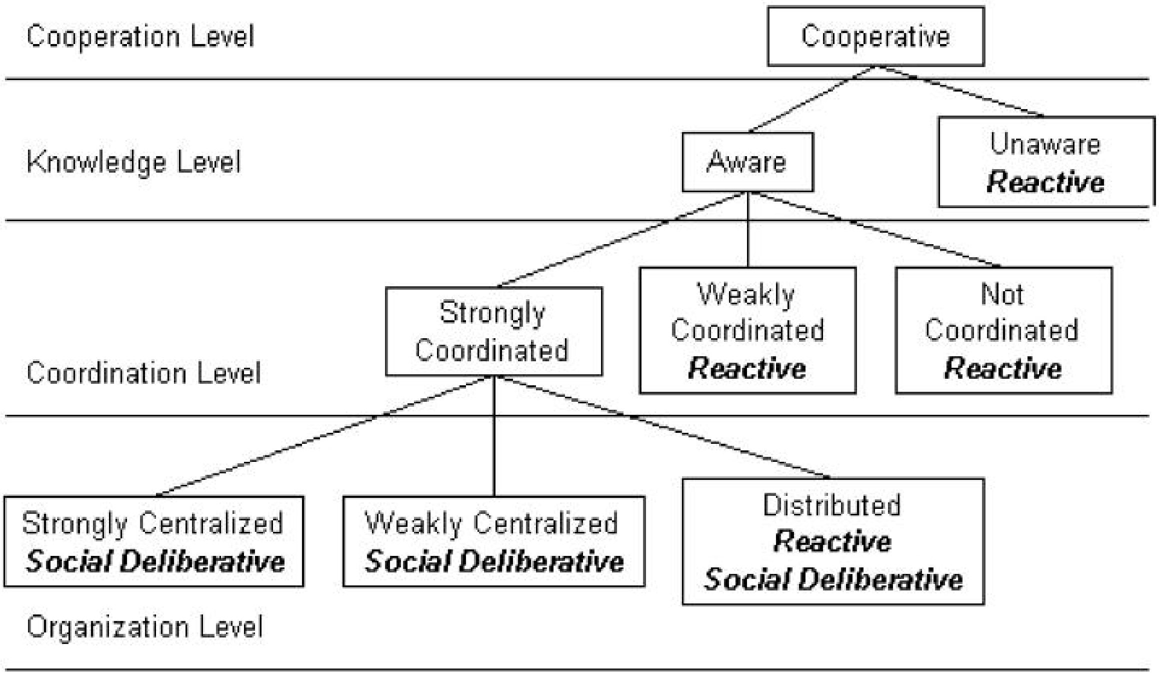

Fig. 1 MRS taxonomy

systems are based on a set of predefined or learned rules concerning in the way two or more robots have to interact. A weak coordinated system does not have any predefined or learned rules on how too interact with other robots.

The organization level

The forth level in the taxonomy is the *Organization Level*. Basically this is the level which decides how the robots work together, weather there is to be an assigned leader robot from the start, *Centralization*, or if the system should be fully autonomous in the decisional process with respect to each other. With other words, no leader is predefined. Centralized systems can also be organized in two sub categories, strong and weak. In a strong centralized system only one robot can act as a leader during the entire match, this is used by some teams predefining the different players as strikers, defenders or goalkeepers. The other category, weak centralized systems allows more than one leader during the match, the leader is assigned dynamically.

Communication and system composition

These are the four pillars in the main taxonomy, but as mentioned earlier two other categories also characterize the MRS, communication and system composition. Communication can be divided into direct- and indirect communication. Direct communication means that the hardware used for communicating with other team members is placed somewhere onboard the robot. Indirect communication is when a robot alters either the environment so the sensory input for another robot changes or when a robot alters the environment so that the effect of another robot's action changes. Direct communication is the preferred choice of communication among the teams in RoboCup, especially in the *humanoid League* and *Middle-Size League*.

The system composition is distinguished into two sub-compositions, *heterogeneous* and *homogeneous*. Robots who are homogeneous are exactly the same both in hardware and in the control software. Heterogeneous robots differentiate themselves in either their hardware or in the control software amongst one another.

L. Iocchi, D. Nardi and M. Salerno define social deliberation and reactivity in their MRS model as follows. MRS social deliberation is a system behavior that allows the team to cope with environmental changes and provides the robots with a strategy so that they can reorganize the team members' tasks. This to use all the resources available to the system and thus reaching the goal as effectively as possible. MRS reactivity is a system behavior where every single robot in the team manages environmental changes by providing a specific solution to reorganize its own task in order to reach the original goal. In the *Middle-Size League* the social deliberation approach has been the more frequent of the two however there are teams that uses the reactivity approach as well.

Robot self-localization

There are many ways for a robot to localize itself in a given world. In RoboCup there are a number of means to help the robot localize different objects on the pitch. For example is color coding used for objects on the pitch, markers are also placed on strategic places on the pitch to help the robots know where they are on the field. In order to do this most teams in the *Middle-Size League* choose to apply a camera to their robot. However filters must be applied in order for the robot to process the information it gets from the camera. There are three main filters used in RoboCup this far and they are *probabilistic Markov localization*, *the Monte Carlo Method* and *particle filtering.*

Background

M. Lauer, S. Lange and M. Riedmiller gives a new approach for robot self-localization with a camera in their article *Calculating the Perfect Match: An Efficient and Accurate Approach for Robot Self-localization* (2005). They recognize three main difficulties with a camera-based self-localization system. They uphold that the self-localization process must be robust. It is difficult to distinguish if objects are outside or inside the pitch since it is not encircled by a board, such objects could for example be spectators. This could lead to misinterpretations of the visual information gathered by the camera. Furthermore position estimation needs to be accurate. Standard camera systems exhibit a poor resolution of objects located more than a few meters away. The pictures taken by the camera can also be misconstrued by the fact that the robots are exposed to recurring accelerations and collisions amongst one another which leads to blurry pictures. The last aspect to take into account according to M. Lauer, S. Lange and M. Riedmiller is that the self-localization approach needs to be computationally efficient because of the need for the robot control program to work in a satisfactory way in strict real time conditions. Three approaches have been made so far in order to solve the self-localization task. As mentioned before, the use of colored landmarks in combination with geometrical calculation. Secondly, detecting white field markers combined with a Hough-transform[1]. The third and last action taken is the detection of landmarks or field markings combined with a sequential importance sampling approach such as *Particle filtering*. M. Lauer, S. Lange and M. Riedmiller claim that all of these approaches have some satisfactory merits but none of them good enough to solve all the objectives of self-localization. They highlight Hough-transform as an example, since this method spends a lot of time calculating coordinates for positions of no interest the computation time increases heavily. They also bring out the Monte Carlo approaches like particle filtering which also wastes a lot of time evaluating positions of no interest. Approximately 98% of positions validated by the particle filtering approach had nothing to do with the final position estimate since positions close to already examined positions are also being evaluated with this approach. M. Lauer, S. Lange and M. Riedmiller therefore want to introduce a new algorithm for robot self-localization which is based on a guided update steps modeling the localization problem as an error minimization task and using an efficient numerical minimizer. Thereafter deriving a measure of reliability of the calculated stochastic sensor fusion process that increases the accuracy of the estimate. This based on the assumptions that the cameras applied on top of the robots are omnidirectional color cameras taking pictures of the environment surrounding the robot and the case of omnidirectional driving capabilities.

The error minimizing algorithm for self-localization

In the image preprocessing state a detector of line points based on an efficient search along pre-defined radial scanlines was used. The function of the preprocessing state is to get a list of positions relative to the robot position and robot heading where scanlines intersect with white field markings previously observed in the image (from here on referred to as *(detected) line points*). In the second state the aim is to find the position and heading of the robot using the information gathered in the preprocessing state. They therefore define an error function that describes the fitness of a certain estimate, this by maximizing (= minimizing the error) the fitness in order to get the best estimate. Assuming the true robot position and heading. The algorithm M. Lauer, S. Lange and M. Riedmiller introduce is as follows.

Let ($\mathbf{p}$, ϕ) be a pair of possible robot positions $\mathbf{p} = (p_x, p_y)$ and heading ϕ in a global coordinate system. The line points list is given relative to the robots pose as vectors $\mathbf{s_1}, \ldots, \mathbf{s}_n$ (see fig x).

[1] In short a Hough-transform is a feature extraction technique used in digital image processing. Originally used to locate lines in pictures, however in recent years it could also be used for locating arbitrary forms.

The world coordinates is therefore given by $\mathbf{p} + \begin{pmatrix} \cos\phi - \sin\phi \\ \sin\phi \cdot \cos\phi \end{pmatrix} \mathbf{s_i}$. Giving the formula for minimizing the error between detected line points and true field markings as follows:

$$\underset{\mathbf{p},\,\phi}{\text{minimize}}\ E :- \sum_{i=1}^{n} \text{err}(d(\mathbf{p} + \begin{pmatrix} \cos\phi - \sin\phi \\ \sin\phi \cdot \cos\phi \end{pmatrix} \mathbf{s_i})) \qquad (1)$$

The mapping d(·) gives the distance from a certain point on the field to the closest field marking. It is continuous and piecewise differentiable and can be calculated from the knowledge of the field markings that are defined in the RoboCup rules.

err is an error function. Since the camera is exposed with a substantial amount of image noise and imperfect image preprocessing the result is compromised with incorrect line points that would distort the estimate. They therefore use $e \rightarrow 1 \; \frac{C^2}{c^2 + e^2}$ with parameter $c \approx 250$, see fig. 2, instead of using the squared error function which is more often used by applications. This function is very similar to the squared one for errors $e \leq c$ and is bounded above by constants for larger errors, thus the influence of outliers onto the estimate is bounded.

Since the minimization problem (1) is non-linear an analytical calculation of the solution cannot be made but the need for a numerical minimizer is still substantial. However *d* is almost everywhere differentiable and can therefore be used to build its gradient in most places and interpolate the gradient at the non-differentiable places. In doing this we can solve the problem with (1). And because of the demand of quick convergence and high robustness 10 iterations of RPROP[2] is used to solve the minimization task.

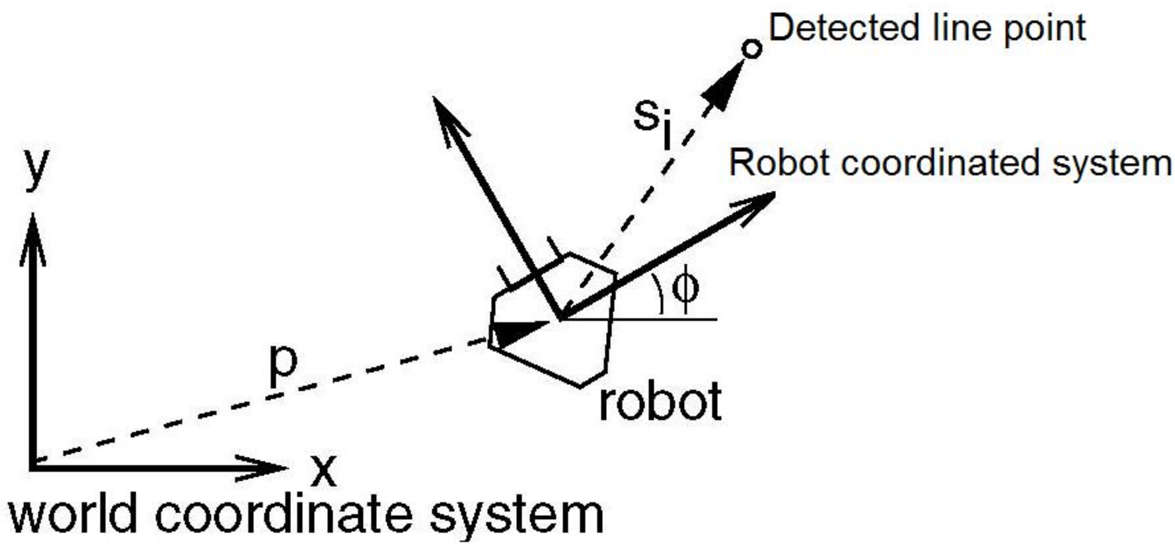

Fig. 2: Sketch of the fixed world coordinate system, the robot relative coordinate system and a vector s_i pointing to a detected line point.

[2] Resilient backpropagation, originally developed as learning rule for multi layer perceptrons but can also be used to solve other types of unconstrained optimization problems.

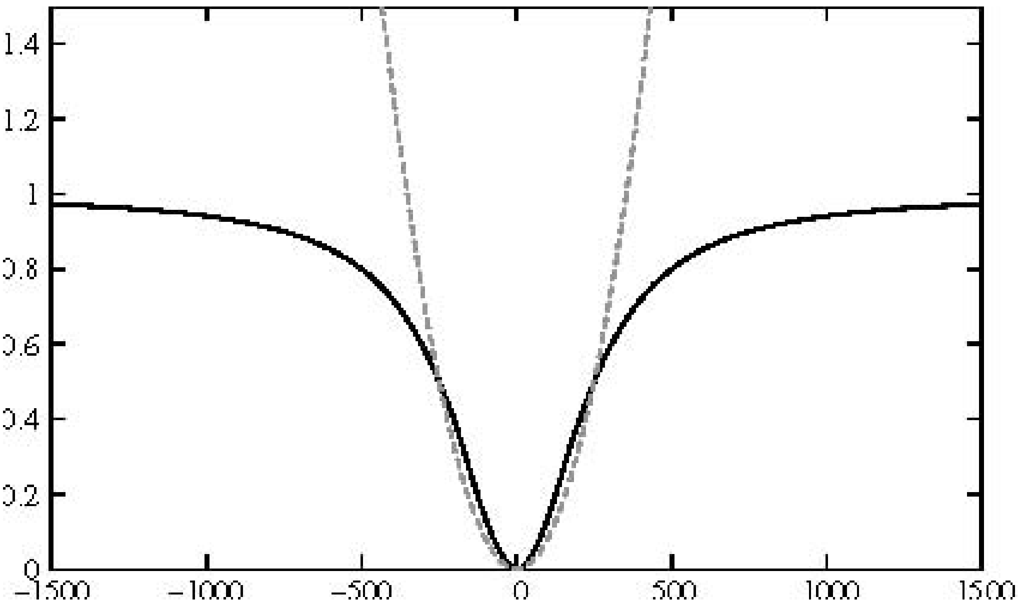

Fig. 3: Comparison of the squared error function (dashed line) and the more robust M-estimator $e \rightarrow 1 - c2/c2{+}e2$ (solid line)

By using the idea of error minimization M. Lauer, S. Lange and M. Riedmiller could post their first draft version of a robot localization algorithm.

"

1. *Start with a known position estimate*
2. *Calculate the movement of the robot since the latest updated position estimate and add it to the latest estimate.*
3. *Optimize the position applying the error minimization approach.*
4. *Reapeat 2. And 3. Every time a new camera image is received.* "

Unfortunately there are a couple of things that restrains the error function. For example a small number of line points could have been gathered or they could be poorly structured. All line points could for example be gathered on the x-axel when a robot is standing next to the touchline. This means that the y-coordinate to the touchline can be estimated very reliably while the x-coordinate still remains vague due to all the data. An aperture problem occurs and in order to deal with this there are three possible situations which have to be recognized. The first one is when the error function exhibits a distinctive global minimum making it possibly to estimate **p** and ϕ reliably. Secondly, such a small number of line points are gathered that the error function becomes completely flat. Resulting in that no parameter can be estimated. The third and last situation is when the error function shows signs of a valley structure around the minimum. In this situation it is possible to estimate parameters robustly if they refer to a coordinate axis orthogonal to the valley however if they are located parallel to the valley an estimation is impossible. This is the situation with the robot standing next to the touchline, hence making it impossible to estimate p_x.

In order to determine the structure of the error function around the minimum M. Lauer, S. Lange and M. Riedmiller propose an analysis of the second order derivates of the error function. This based on the case where the value of $\partial^2 E/(\partial p_x)^2$ is small and parallel to the x-axis and in the completely flat case while it is large if *E* shows a distinctive minimum with regards to the x-axis. This resulting in that it is now possible to analogously analyze $\partial^2 E/(\partial p_y)^2$ and $\partial^2 E/(\partial \phi)^2$.

Since the function *d* is built upon the line markings on the field and most of them (all but the centre circle and corner arcs) are parallel to the coordinate axis. This means that the *d* function is piecewise linear in most parts of the field giving us second order derivates which are zero in these areas, which in turn means that we can simplify the function accordingly:

$$\partial^2 E/(\partial p_x)^2 \approx \sum_{i=1}^{n} err''(s_i) \cdot (\partial d(s_i)/\partial x)^2 \qquad (2)$$

$$\partial^2 E/(\partial p_y)^2 \approx \sum_{i=1}^{n} err''(s_i) \cdot (\partial d(s_i)/\partial y)^2 \qquad (3)$$

$$\partial^2 E/(\partial \phi)^2 \approx \sum_{i=1}^{n} \Big(err''(s_i) \cdot (\partial d(s_i)/\partial x)^2 (-\sin\phi - \cos\phi) s_i + \partial d(s_i)/\partial y (\cos\phi - \sin\phi) s_i)^2$$

$$+ err' \cdot \big(\partial d(s_i)/\partial x (-\cos\phi \sin\phi) s_i + \partial d(s_i)/\partial y (-\sin\phi - \cos\phi) s_i\big)\Big) \qquad (4)$$

where err' and err'' denote the first and second order derivative or err.

M. Lauer, S. Lange and M. Riedmiller present one unfortunate problem with the error function $e \rightarrow 1 \frac{C^2}{c^2 + e^2}$ used in (1), it is not completely positive definite and therefore the curvature criterion may be misleading. This means that the second order partial derivative may be small also when the minimum is distinctive. In order to avoid the problem they adopt the following artifact: in (2) – (4) the original error function is replaced by the squared error function $e \rightarrow ½(e/c)^2$ and outlying observations are being ignored. *E* now becomes positive definite and the results produced by the curvature criterion becomes satisfactory again.

Because of movement and collisions on the field the images taken by the camera the pictures could be exposed to severe noise and inaccuracy damage, causing the algorithm to calculate false values. To reduce the noise in the pictures an evaluation of the temporal dependency of positions estimated from subsequent images is proposed. Since these are neighbored and linked using some transition depending on the velocity a stochastic weighted averaging approach, it is a simplified application of the Kalman filter. All estimates with variances that model the degree of uncertainty are being enclosed and no covariance's is used to simplify the modeling.

Let denote ($\mathbf{r}_t$, ψ_t) the estimate of the robot's position and heading at time t and $\sigma^2_{r_x,t}$, $\sigma^2_{r_y,t}$, $\sigma^2_{\psi,t}$ the respective variances. After a robot movement position estimate and variances are updated using the motion model of an omnidirectional robot with velocity $\boldsymbol{\upsilon}$ and rotational velocity[3] ω:

$$\hat{\psi}_{t+\tau} = \psi_t + \omega \cdot \tau \qquad (5)$$

$$\hat{\mathbf{r}}_{t+\tau} = \begin{cases} \mathbf{r}_t + \boldsymbol{\upsilon} \cdot \tau & \text{if } \omega = 0 \\ \mathbf{r}_t + (1/\omega) R_{\psi t} \begin{pmatrix} 1 - \cos(\omega\tau) \cdot \sin(\omega\tau) \end{pmatrix} R_{-\psi t}\, \boldsymbol{\upsilon} & \text{if } \omega \neq 0 \end{cases} \qquad (6)$$

with R_ψ denoting the rotation matrix by the angle ψ. Odometers are used to measure the velocity and rotational velocity. The update of the variances takes into account the inaccuracy of the movement:

$$\sigma^2_{\hat{\psi},t+\tau} = \sigma^2_{\psi,t} + \alpha(\hat{\psi}_{t+t} - \psi_t)^2 \qquad (7)$$

$$\sigma^2_{\hat{r}_x,t+\tau} = \sigma^2_{r_x,t} + \alpha(\hat{r}_{x,t+t} - r_{x,t})^2 \qquad (8)$$

$$\sigma^2_{\hat{r}_y,t+\tau} = \sigma^2_{r_y,t} + \alpha(\hat{r}_{y,\, t+t} - r_{y,}\, t)^2 \qquad (9)$$

The parameter $\alpha > 0$ controls the assumed accuracy of the movement.

In (8) and (9) the non-linear dependency between rotational and translational movements of a robot is ignored. By doing this the ability to easily manipulate the statistical modeling efficiently remains high. As long as the update rate of the images remains high, the additional error made by the assumption of independence stays small.

After receiving and calculating the optimal estimate with respect to the image information ($\mathbf{p}$, ϕ) it is possible to calculate a smoothed position estimate combining ($\mathbf{p}$, ϕ) and ($\mathbf{r}$, ψ). Therefore they introduce variances for ($\mathbf{p}$, ϕ) that model the uncertainty of the image-based estimator.

The reliability of the image-based estimator relies on a number of aspects such as, mechanical vibrations, camera calibration points, preprocessing accuracy and the structure of detected line points. It is important to model the structure of the line points so that any erroneous estimates are avoided. To do this curvature criterion is used to individually determine the variance of each parameter: a small second order partial derivative should be related to a large variance and the other way around. M. Lauer, S. Lange and M. Riedmiller used a heuristic function

[3] $\boldsymbol{\upsilon}$ and ω refer to the global coordinate system and describe the movement at time t.

produced through a set of experiments by visual inspection to map second order partial derivatives onto variances.

The sensor fusion step takes two independent Gaussian distributions. Denoting with σ^2 the variances of ϕ:

$$\psi_{t+\tau} = \frac{\sigma_\phi^2 \hat{\psi}_{t+\tau} + \sigma_{\hat{\psi},t+\tau}^2 \phi}{\sigma_\phi^2 + \sigma_{\hat{\psi},t+\tau}^2} \tag{10}$$

$$\sigma_{\psi,t+\tau}^2 = \frac{\sigma_\phi^2 \cdot \sigma_{\hat{\psi},t+\tau}^2}{\sigma_\phi^2 + \sigma_{\hat{\psi},t+\tau}^2} \tag{11}$$

The sensor fusion steps for $\mathbf{r}_{t+\tau}$ can be calculated analogously.

When using the filtered estimates both robustness and precision improves since one misleading image does not lead to a change in direction for the robot because this filter does not allow it.

This algorithm is a further step towards being a completely autonomous and robust soccer playing robot; its computation time is far less than other algorithms used in self-localization purposes e.g. it has ten time faster computational time than the particle filter. The algorithm itself is characterized by three properties: high accuracy, robustness and efficiency.

Conclusion

Both the MRS and the error minimizing algorithm are exciting tools used to make robots more autonomous. But I must remain skeptical against the goal of RoboCup, I do not think that by 2050 a team of robots are able to beat the best nation in football in the world at that moment. Not only do I believe that humans have and will have better agility and flexibility at that time than robots. I also believe humans will have a different feel for the ball on the pitch, a sense of ball-foot coordination which I believe will be very hard to implement to robots.

In this article I have taken a look at the communication between robots and different means of getting robots to both be aware of each other and to communicate with each other. Something that I believe is a very important area of research, not only to create good football-playing robots but for example in search and rescue robots. The different ways to define a leader and the quantity of information every robot gets is important, especially if a malfunction occurs and one of the robots shuts down. Not preferred if people need fast extractions during earthquakes for example. The same thing applies for the self-localization algorithms, the better they get the better robots get. And if they get good enough who knows what they could be used for.

To conclude I have to say that RoboCup is an important competition in the development of robots and robotic systems. It is a fun way to take the research forward and it is a smart way getting the multinational corporations to get involved and finance the research. Who knows, one day we might have a football league consisting of robots.

References

1. Martin Lauer, Sascha Lange and Martin Riedmiller (2005). *Calculating the Perfect Match: An Efficient and Accurate Approach for Robot Self-localization*, Springer-Verlag Berlin Heidelberg

2. Luca Iocchi, Daniele Nardi and Massimiliano Salerno (2001). *Reactivity and Deliberation: A Survey on Multi-Robot Systems*, Springer-Verlag Berlin Heidelberg

3. Sven Behnke, Jürgen Müller and Michael Schreiber (2005). *Toni: A Soccer Playing Humanoid Robot*, Springer-Verlag Berlin Heidelberg

4. Sven Behnke, Jürgen Müller and Michael Schreiber (2005). *Playing Soccer with RoboSapien*, Springer-Verlag Berlin Heidelberg

5. Richard O. Duda and Peter E. Heart (1971), *Use of the Hough Transformation to Detect Lines and Curves In Pictures*, Artificial Intelligence Center, Comm. ACM, Vol 15, No. 1, pp. 11-15