Tracking Dynamic Objects in a RoboCup Environment - The Attempto Tübingen Robot Soccer Team

Patrick Heinemann, Michael Plagge, André Treptow, and Andreas Zell

Wilhelm-Schickard-Institute, Department of Computer Architecture,
University of Tübingen, Sand 1, 72076 Tübingen, Germany
{heineman, plagge, treptow, zell}@informatik.uni-tuebingen.de

Abstract. This paper describes the Attempto Tübingen Robot Soccer Team which played at RoboCup 2003 in Padova. The robot platform, its sensors and actuators, and the software system running on the onboard computer are presented. The main part of the paper concentrates on our current scientific work on modelling and tracking a dynamic environment. Information about dynamic objects moving around in the environment can be useful especially in RoboCup to predict the motion of the ball, to avoid collisions, or to consider objects which can not be detected over a short period of time. In our robot soccer team we implemented an object tracking approach which on the one hand combines the specific advantages of Kalman- and particle filters and on the other hand uses an interacting multiple model filtering approach to model object dynamics as accurately as possible. In addition to the general tracking techniques we present a new real-time approach to detect and track uncoloured objects, such as a standard soccer ball.

1 Introduction

Teams of cooperative robots for solving a given task are often based on the idea of a highly precise sensor system giving the robot a complete and accurate view of its environment. In our RoboCup Middle Size League team we followed this idea by building our robots around a very precise laser scanner, at the cost of loosening other constraints like small size and light weight. Although we still played with the laser scanner at RoboCup 2003 at Padova, several aspects including recent rule changes force us to remove that sensor from our robots. In this paper we will present our current team of robots and our reconstruction plans for being competitive in future events. We believe that our scientific research on tracking dynamic objects will help us to cope with imperfect and incomplete sensor data. In any case we are prepared for future rule changes concerning the orange ball through our new real-time approach to track uncoloured objects, such as a standard FIFA soccer ball. The remainder of this paper is structured as follows: Section 2 briefly describes the robot, sensor and computer hardware of the Attempto robots, whereas Section 3 focuses on the software controlling the robots. The main part of the paper, however, deals with our current scientific research on object tracking in Section 4. Section 5 concludes the paper with a short summary.

2 Hardware

2.1 Robot Platform

We are currently using the Pioneer2 DX platform from ActivMedia Inc. as the basic robot platform for our field players. This platform is driven by a differential drive system which can achieve translational speeds up to 1.5 m/s and rotational speeds up to 2π/s. Due to the heavy load the robots were carrying at Padova (cf. section 2.2) we reduced the maximum translational speed to 1.0 m/s. The Pioneer2 DX can carry weights up to 20 kg and can be equipped with a maximum of three 7,2 Ah batteries, which allow an operating time of nearly three hours including all additional hardware like onboard PC and additional sensors and actuators.

The two driving motors are equipped with 500 tick position encoders. With the data from these encoders the speed and the relative position and orientation of the robot can be calculated by the onboard Siemens C166 microcontroller board which is also responsible for controlling the actuators of the robot.

The communication between the controller board and a remote computer is done via a RS232 serial connection at a maximum speed of 38400 baud. The controller sends status data packets to the remote computer at a rate of 20 Hz and accepts commands from it at the same rate. Therefore the minimal achievable response time for a closed loop controller is about 50 ms. To reduce this time delay and to get a higher precision of odometry we are currently working on a custom designed Motorola MC68332 CPU board to replace the old C166 board. This board already proved to be useful in our goalkeeper in RoboCup 1998 but was never transfered to the field players.

Our goalkeeper at Padova was based on an old Pioneer AT platform by ActivMedia Inc. The skid steering drive used by this platform involves several problems when used as a goalkeeper. This type of drive is suitable for a goalkeeper that stays on the goal line but as the speed of robots and the ball rises there is a need for a more agile platform, especially for the goalkeeper. Therefore we plan to replace the Pioneer AT platform of our goalkeeper with an omnidirectional platform developed by the University of Dortmund for their own robot team ([1]).

2.2 Sensors and Actuators

In the past we were employing a diversity of sensors, being convinced that the use of several sensors can result in a highly redundant system and, by the use of suitable data fusion techniques, a better assessment of the situation around the robot. A maximum of six different sensor types including sonars, a 2D laser range finder, two different types of vision systems, infrared proximity sensors, and a digital compass was used on our robots in several configurations. However, the constantly changing environment in RoboCup reduced the applicability of several sensors while others, such as the sonars, and infrared proximity sensors where simply outperformed by better sensors like the high accuracy laser scanner. This led to a reduced number of three sensor types during RoboCup, the laser scanner, an omnidirectional vision system, and a standard perspective camera pointing forward which are described in this section. The trend towards a small number of sensors is further pushed by the need of fast and reactive robots that are able to handle a ball shot by the new kicking devices that are able to accelerate the ball to several metres per second. Therefore we are thinking about removing our heavy laser scanner to get lighter and faster robots.

Apart from the motors we have our robots equipped with only one more actuator. This pneumatic kicking device will be described in this section, too.

Laser Scanner: The LMS200 laser measurement system by SICK AG was our main sensor for precise object localisation and self-localisation in the past. Its characteristics are a 180° field of view, an angular resolution of 1°, 0.5°, or even 0.25° (software programmable), and an accuracy of 10 mm up to a distance of 10 m. With a resolution of 1° and a 500 kbps data transfer rate over a highspeed RS422 serial device it is possible to achieve a scan rate of nearly 40 Hz. However, the main drawbacks of this sensor are its size (137x156x185 mm), weight (4.5 kg), and power consumption (max. 17.5 W).

We had achieved a very accurate self-localisation based on scan matching methods when there were walls around the field ([10]). We had some problems with these methods when the walls were replaced by a row of posts around the field, but we were still able to localise based on laser scans. For RoboCup 2003, however, we had to implement an entirely new self-localisation method based on our omnidirectional vision systems, as all borders around the field were removed except a security border, below the sensing half-plane of the laser scanner.

With this new situation the laser scanner was reduced to recognise objects on the field at a very high and sometimes unnecessary accuracy. Therefore, and keeping the drawbacks as heavy weight and the power consumption in mind, we now decided to replace the object recognition by the omnidirectional vision system, too. Doing so we could remove the laser from the robot and undo the speed limitation we introduced to protect the motor controllers from overheating.

Cameras: The two vision systems we have installed on the robot (perspective camera and omnidirectional vision system) both use a Siemens SICOLOR C810 CCD-DSP color camera with a 1/3 inch CCD chip and a resolution of 752x582 pixels. The output of the camera is a regular CCIR-PAL signal with 625 rows and 50 half frames per second.

One of the cameras is equipped with a 2,8f wide angle lens and is mounted at the front of the robot. It is used for a precise detection of the ball.

The second camera is equipped with a 4,2f lens and is placed on top of the robot pointing upwards. A hyperbolic mirror of the Fraunhofer Gesellschaft (FhG-AiS) is mounted above the camera enabling the vision system to get a mapping of the complete surrounding of the robot up to the horizon. Although the mirror was designed for a vision system on top of the FhG Volksbot ([2]) we are achieving good results with our system, too.

Kicker: Our kicking device is a self-made construction actuated by compressed air. The air is compressed into a 2 litre tank before the games at a pressure of 10 bar. The air reservoir is connected via an electric valve to two pneumatic actuators that can accelerate a metallic bar which shoots the ball. The special feature of this device compared to others is that the bar is mounted and connected to the pnematic cylinders in a way that accelerate the bar in a circular motion forwards and also upwards. This reduces the overall speed of the ball but leaves the possibility to lift the ball over a goalkeeper as we could show in a game against ISePorto at RoboCup 2003 (see videos on [3]).

2.3 Onboard Computer

Our onboard computer is a custom designed system based on a PISA-3P4I Backplane by JUMPtec which provides 4 ISA/PISA slots and 3 PCI slots. One of these slots is used to plug a CoolMonster/P3 PISA board by Jumptec which integrates the complete functionality of a motherboard, like a network controller, IDE controller, and USB controller. This board is equipped with a low power Pentium-3 running at 850 MHz, 128 MB of RAM and a 20 GB harddisk. Additionally two PCI framegrabber boards based on the Booktree BT848 chipset are added to the computer to simultaneously grab the images of the two vision systems at 25 fps. The laser scanner is connected via a high speed RS422 serial device card which was modified to achieve the 500 kbps data rate. The computers of different robots can communicate via IEEE 802.11b wireless LAN by ARtem Datentechnik over an external access point. Therefore each robot is equipped with a WLAN client which is connected to the onboard computer via RJ45 network cable. The communication to the controller board of the robot is done over the RS232 serial device and a crosslink cable. We are running RedHat 7.3 Linux on the computer.

3 Software

The software system of the Attempto Tübingen Robot Soccer Team is based on a Client/Server architecture and can be divided into three layers: the data server layer, an intermediate layer and the high level robot control layer.

In the data server layer several server programs perform the communication with the sensor and robot hardware. They provide the data from the sensors and the robot to the preprocessing clients in the intermediate layer via shared memory segments. These segments are organised in a ring buffer structure to provide a free buffer for the next data packet even if one or more clients are processing data from other segments and thus blocking the use of these segments. The robot server that supplies odometry data is actually a client, too, as it reads command data from a shared memory segment and makes the robot fulfill these commands. All the servers in this layer can be replaced by simulation servers which provide previously recorded or artificial data for simulation purposes.

The intermediate layer acts as a data compression layer. Several preprocessing stages extract the relevant information out of the raw sensor data and provide it to the high level layer, again being both client and server. The image preprocessing stage computes the position of objects (own robots, opponent robots, and the ball) relative to the robot and extract points on the white field markings. The laser preprocesing stage extracts objects and line segments from the laser scan. In an object tracking stage the objects generated from the image and the laser preprocessing stage are fused to further reduce the amount of data and to remove inconsistencies and the remaining objects are tracked over time by our tracking system presented in section 4.1. A localisation stage processes the line segments from the laser scan, the field markings from the images, and the odometry data from the robot to generate new position estimations. These estimations are used to update a filter that keeps track of the robot's position. The output of the stages in the intermediate layer provide a consistent world model to the high level layer.

The high level robot control layer realises the hybrid robot control architecture [5]. It consists of a reactive component where a set of independent behaviours like obstacle avoidance, ball search, or ball following try to fulfill their tasks. The behaviours can react quite fast to changes in the environment because they can work on the compact world model data from the immediate layer. The behavioural system is easy to expand beacuse it is possible to start and stop behaviours at runtime. Control commands are passed to the robot via the robot server. A detailed description of the software system is given in [10].

4 Research Topics

4.1 Tracking Dynamic Objects

Modelling the environment of autonomous mobile robots has proven to be very useful for planning tasks and robot control. Early approaches concentrated on modelling the static environment by map building and self localization within such global maps. Recent research also tries to model the dynamic objects moving around in the environment. Most of this work concentrates on the aspect of tracking multiple targets. Different techniques of modelling like multiple hypothesis tracking (MHT) [12] or joint probabilistic data association filters (JPDAF) and their extensions to particle filters [13], among others [8], were suggested. Here we will concentrate on a sophisticated single object tracker based on an interacting multiple model (IMM) filtering approach first proposed by Blom [6].

Tracking a maneuvering target with a filter which utilises a single dynamic model for target motion like a single Kalman filter or a particle filter usually involves artificial high process noise. This noise is used for compensating target movements which are not complying with the dynamic model and deteriorates the filter outcome. Additionally the prediction of the further target track for planning purpose is of little value, if the target motion is not governed by the assumed dynamic model.

So if tracking a maneuvering target with a single dynamic model has the mentioned drawbacks, it is obvious to try to use an approach which utilises a set of dynamic models for target tracking. Consider such a set of s models, where at each point in time at least one model i is appropriate to describe the target movement. At discrete points k in time a certain probability exists for the target to switch the mode of movement, which means at time k a different model j characterizes the target movement. Although the mathematical structure of the optimal solution of such multi model tracking problems is well understood, it is practically not tractable, since the complexity of the solution is increasing exponentially in time. The reason for this exponential behavior is the increasing number of possible model sequences. For each single model sequence at time k there are s successors under the assumption that the transition probability from one model to another is not zero. As a result various suboptimal approximations have been proposed. One of these suboptimal algorithms that is widely used for state estimation in target tracking is the interacting multiple model (IMM) algorithm, which we will describe next.

With the restriction on linear dynamic models, the multiple model tracking approach can formally be seen as a jump Markov linear system. Consider the following system, which can be in one of s modes with r_k identifying the mode of the system at time k:

$$x_{k+1} = F_k^{r_{k+1}} x_k + B_k^{r_{k+1}} v_k \tag{1}$$

$$y_k = H_k^{r_k} x_k + w_k \tag{2}$$

where v_k and w_k are independent white Gaussian noise processes with covariance $Q_k^{r_k}$ and $R_k^{r_k}$ respectively. The matrix functions $F_k^{r_k}, B_k^{r_k}, H_k^{r_k}$ are assumed to be known. r_k is modeled as a discrete, first order Markov chain with s modes. Let $S = \{1, 2, \ldots, s\}$. The transition probability from mode i to mode j given by:

$$p_{ij} = Pr\{r_{k+1} = j | r_k = i\} \quad \text{and} \quad 0 \leq p_{ij} \leq 1 \quad i, j \in S, \quad \sum_{j=1}^{s} p_{ij} = 1 \tag{3}$$

is also assumed to be known. The initial state distribution of the Markov chain is $\pi = [\pi_1, \ldots, \pi_s]$ where

$$0 \leq \pi_j \leq 1, \quad j \in S, \quad \sum_{j=1}^{s} \pi_j = 1 \tag{4}$$

Now our aim is to recursively estimate the state x_k of the system from a sequence of given measurements $\{y_k, k \in \mathbb{N}\}$. For this we apply the IMM algorithm consisting of a bank of Kalman filters, each representing a certain mode of target movement and a logic to combine the filtered outcome for a target state estimate. In the following $\hat{x}_{m|n}$ denotes the target state estimate at time m conditioned on all measurements y_k up to time n. $P_{m|n}$ is the associated covariance matrix. Quantities relevant to mode or Kalman filter j are denoted with superscript j. Hence $\hat{x}^j_{m|n}$ and $P^j_{m|n}$ describe the state estimate and covariance matrix of Kalman filter j respectively. $\hat{x}^{0j}_{k-1|k-1}$ and $P^{0j}_{k-1|k-1}$ are the mixed prior for the same filter j. μ^j_k is the probability for Kalman filter j to match model movement at time k. $N(x, \bar{x}, P)$ refers to the Gaussian density function of x with mean $\bar{x}$ and covariance matrix P. Apart from the initialization step the IMM consists of three steps, which are recursively applied.

Initialization:

$$\mu_0(j) = \pi_j, \quad \forall j \in S, \quad , \hat{x}_0 \sim N(0, P_0) \tag{5}$$

Recursion:

Step 1) Filter input calculation

$$\forall i, j \in S \quad \mu_{k-1}(i|j) = \frac{1}{\bar{c}_j} p_{ij} \mu_{k-1}(i) \quad \text{with} \quad \bar{c}_j = \sum_i p_{ij} \mu_{k-1}(i) \tag{6}$$

$$\hat{x}^{0j}_{k-1|k-1} = \sum_{i \in S} \mu_{k-1}(i|j) \hat{x}^i_{k-1|k-1} \tag{7}$$

$$P^{0j}_{k-1|k-1} = \sum_i \mu_{k-1}(i|j) \{ P^i_{k-1|k-1} + [\hat{x}^i_{k-1|k-1} - \hat{x}^{0j}_{k-1|k-1}][\hat{x}^i_{k-1|k-1} - \hat{x}^{0j}_{k-1|k-1}]^T \} \tag{8}$$

Step 2) Kalman filtering

Kalman filter time prediction

$$\hat{x}^j_{k|k-1} = F^j_k \hat{x}^{0j}_{k-1|k-1} + B^{r_{k+1}}_k \bar{v}_k \tag{9}$$

$$P^j_{k|k-1} = F^j_k P^{0j}_{k-1|k-1} {F^j_k}^T + Q^j_k \tag{10}$$

Kalman filter measurement update

$$\hat{x}^j_{k|k} = \hat{x}^j_{k|k-1} + K^j_k r^j_k \tag{11}$$

$$r^j_k = y_k - H^j_k \hat{x}^j_{k|k-1} \qquad \text{(residual)} \tag{12}$$

$$K^j_k = P^j_{k|k-1} {H^j_k}^T {S^j_k}^{-1} \qquad \text{(kalman gain matrix)} \tag{13}$$

$$S^j_k = H^j_k P_{k|k-1} {H^j_k}^T + R^j_k \qquad \text{(residual covariance)} \tag{14}$$

mode probability update

$$\Lambda^j_k = N(r^j_k, 0, S^j_k) \qquad \text{(likelihood function)} \tag{15}$$

$$\mu_k(j) = \frac{1}{c} \Lambda^j_k \sum_i p_{ij} \mu_{k-1}(i) = \frac{1}{c} \Lambda^j_k \bar{c}_j \tag{16}$$

Step 3: Output combination

$$\hat{x}_{k|k} = \sum_j \hat{x}^j_{k|k} \mu_k(j) \tag{17}$$

$$P_{k|k} = \sum_j \mu_k(j) \{ P^j_{k|k} + [\hat{x}^j_{k|k} - \hat{x}_{k|k}][\hat{x}^j_{k|k} - \hat{x}_{k|k}]^T \} \tag{18}$$

In the concrete case of tracking robots in a robocup environment we are using two different ($s = 2$) models for the target dynamics. The first is the following constant velocity model, in which target acceleration is modeled as white noise. The states of the filter are given by the target position (x, y) and velocity (v_x, v_y), $\hat{x} = [x, v_x, y, v_y]$. The complete model according to (1) is given by:

$$\hat{x}_{k+1} = \begin{pmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{pmatrix} \hat{x}_k + \begin{pmatrix} \frac{T^3}{3} & \frac{T^2}{2} & 0 & 0 \\ \frac{T^2}{2} & T & 0 & 0 \\ 0 & 0 & \frac{T^3}{3} & \frac{T^2}{2} \\ 0 & 0 & \frac{T^2}{2} & T \end{pmatrix} \tag{19}$$

where T denotes the elapsed time between $k + 1$ and k. This model matches robot movements along a straight line with acceleration parallel to the movement direction.

The second model denotes a robot performing a turn. The states of this filter are given also by the target position (x, y) and velocity (v_x, v_y). Additionally the target turning rate ω is modeled, $\hat{x} = [x, v_x, y, v_y, \omega]$. The nonlinear target dynamics $\hat{x}(k + 1) = f(\hat{x}(k), v(k))$ is given by:

$$x(k+1) = x(k) + \frac{\sin(\omega T) v_x(k)}{\omega} - \frac{1 - \cos(\omega T) v_y(k)}{\omega}) \tag{20}$$

$$y(k+1) = y(k) + \frac{1 - \cos(\omega T) v_x(k)}{\omega}) + \frac{\sin(\omega T) v_y(k)}{\omega} \tag{21}$$

$$v_x(k+1) = v_x(k) \cos(\omega T) - v_y(k) \sin(\omega T) \tag{22}$$

$$v_y(k+1) = v_x(k) \sin(\omega T) + v_y(k) \cos(\omega T) \tag{23}$$

$$\omega(k+1) = \omega(k) \tag{24}$$

Since the target dynamics is nonlinear we are using an Extended Kalman filter (EKF). Therefore the Kalman filter equations of the second step in IMM must be replaced by the appropriate equations from the EKF. We plan to integrate these filtering techniques in our system to satisfy the need for a better compensation of imperfect and noisy data coming from an image processing system in contrast to a precise laser measurement system. Preliminary tests have already shown the applicability to the RoboCup environment.

4.2 Tracking Uncoloured Objects

In the RoboCup environment every object is marked with a special colour so that fast and robust colour segmentation algorithms can be used for object detection

[7][14]. In the future these colour markers will be removed in order to come to a more realistic setup. Therefore, the aim of our research is to introduce algorithms to detect and track objects that do not have colour information. In a first step we want to be able to replace the orange soccer ball and play with a standard FIFA ball. In this section we will give a short overview of this work. A more detailed description can be found in [4].
To build a colourless model of the ball we use an algorithm proposed by Viola and Jones [9] that has been used to detect faces in real-time based on simple gray-level features. We used their approach to come to a feature based description of the ball. As proposed in [9] we use four different types of features (see figure 1).

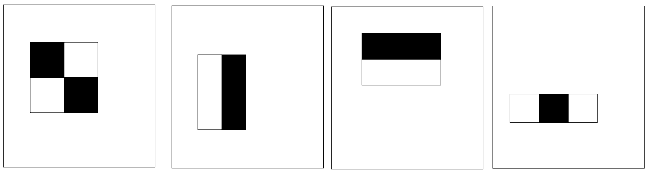

Fig. 1. Four different types of rectangle features within their bounding box. The sum of pixels in the white boxes are subtracted from the sum of pixels in the black areas.

The advantage of using these features is that they can be calculated very fast on top of a so called integral image (see [9] for details). To build a ball classifier based on these gray level features one has to select the most relevant features. Remember that within a box sized 24x24 there are more than 160000 features, which is far more than the number of pixels. As proposed by Viola *et. al.*, we use the machine learning procedure called Adaboost [15] to select a small number of relevant features. For the offline training of the ball classifier we collected a set of 1100 pictures (sized 19x19) showing the ball under different viewing conditions and 10000 pictures that do not contain the ball. These sets are randomly split into a training and a test set. To classify the training set correctly, Adaboost selects 137 features. On the test set we achieve a detection rate of 91.64% and a false positive rate of 0.015%.
To track the ball we use a particle filter: The ball is described by the state vector

$$x_t^{(i)} = [x_I, y_I, s_I, v_x, v_y, v_s]^T \tag{25}$$

where (x_I, y_I) is the position of the ball in the image, (v_x, v_y) is the velocity in x- an y-directions, s_I represents the size of the ball and v_s is the velocity in size. The dynamics of the ball are modelled as a movement with constant velocity and small random changes in velocity (random walk). Every particle is weighted with the classification result of the ball classifier that has been learned offline by the Adaboost mechanism. Instead of using the binary value we weight every particle with the result of the linear combination of the features. We use random initialization for the particle filter.

Using 300 samples and a ball classifier with 40 features, one timestep of the tracking algorithm requires about 30ms on a Pentium III 850MHz processor so that we are able to track the ball with more than 25 fps. In different experiments the tracker has shown to be robust against occlusion and distraction. Examples of our ball tracking system can be seen in figure 2 and 3.

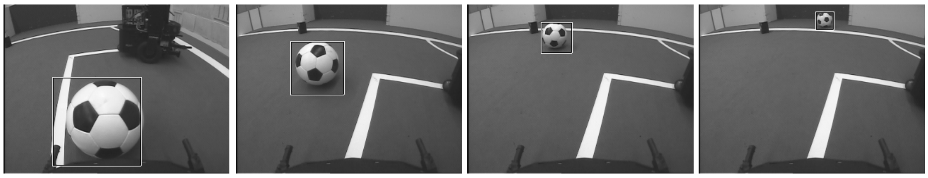

Fig. 2. Tracking the ball at different scales.

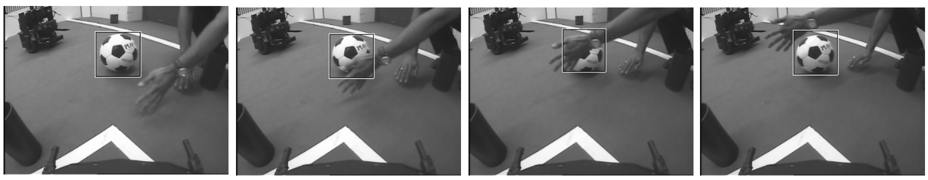

Fig. 3. Tracking the ball through occlusion.

We treat the weighted mean of the best 30% of the particles to be the final hypothesis of the ball position. Nevertheless, there are situations were we get false detections so that the tracker is not able to recover from distraction. To improve robustness further, we will implement methods to measure the shape. Besides the work of Hanek *et. al.* [11] who do not deal with the problem of global detection, our approach is one of the first to detect and track a "normal", non-coloured FIFA-ball in real-time. The presentation of this work was the main reason for winning the Technical Challenge Award of the Middle Size League at the RoboCup world championship in Padova 2003.

5 Summary and Diskussion

In this paper we introduced our current and new approaches to stay competitive in future RoboCup competitions. Besides the need to reduce weight to get faster and more flexible robots, we mainly focus on research to improve the capabilities of our team. We believe that our ideas on object tracking with IMM filtering techniques will enhance the internal representation of the environment and thus the overall performance of our robots. First tests with previously recorded data seem to confirm this but we still have to prove the results in our robots. Being able to detect and track a standard uncoloured FIFA ball in real-time as one of the first teams in the world we are also prepared for a further reduction of the colour markings on the field. In Padova 2003 we could successfully present this new ability.

References

1. http://ls1-www.cs.uni-dortmund.de/~pg425/.
2. http://www.volksbot.de.
3. http://www-ra.informatik.uni-tuebingen.de/forschung/robocup/videos_e.html.
4. A. Treptow, A. Masselli and A. Zell. Real-Time Object Tracking for Soccer Robots without Color Information. In *Proceedings of the European Conference on Mobile Robots (ECMR 03)*, September 2003.
5. Ronald C. Arkin. *Behavior-Based Robotics*. The MIT Press, 1998.
6. H.A.P. Blom. An efficient filter for abruptly changing systems. In *IEEE Conference on Decision and Control*, 1984.
7. J. Bruce, T. Balch and M. Veloso. Fast and inexpensive color image segmentation for interactive robots. In *Proc. of the 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '00)*, volume 3, pages 2061–2066, 2000.
8. M. Montemerlo, W. Whittaker, and S. Thrun. Conditional particle filters for simultaneous mobile robot localization and people-tracking. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 1, pages 695 –701, 2002.
9. P. Viola and M.J. Jones. Robust real-time object detection. In *Proc. of IEEE Workshop on Statistical and Theories of Computer Vision*, 2001.
10. M. Plagge, R. Günther, J. Ihlenburg, D. Jung, and A. Zell. The attempto robocup robot team. In H. Kitano M. Veloso, E. Pagello, editor, *RoboCup-99: Robot Soccer World Cup III*, volume 1856 of *Lecture Notes in Artificial Intelligence*, pages 424–433. Springer Verlag, 2000.
11. R. Hanek, T. Schmitt, S. Buck and M. Beetz. Towards RoboCup without Color Labeling. In *RoboCup International Symposium*, Fukuoka, Japan, 2002.
12. T. Schmitt, R. Hanek, M. Beetz, and S. Buck. Watch their moves: Applying probabilistic multiple object tracking to autonomous robot soccer. In *The Eighteenth National Conference on Artificial Intelligence*, 2002.
13. Dirk Schulz, Wolfram Burgard, Dieter Fox, and Armin B. Cremers. Tracking multiple moving targets with a mobile robot using particle filters and statistical data association. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1665–1670, 2001.
14. T. Bandlow, M. Klupsch, R. Haneka and T. Schmitt. Fast Image Segmentation, Object Recognition and Localization in a RoboCup Scenario. In *3. RoboCup Workshop, IJCAI'99*, 1999.
15. Y. Freund and R.E. Schapire. A short introduction to boosting. *Journal of Japanese Society for Artificial Intelligence*, 14(5):771–780, September 1999.