

Visual Robot Detection in RoboCup Using Neural Networks

Ulrich Kaufmann, Gerd Mayer, Gerhard Kraetzschmar, and Günther Palm

University of Ulm,
Department of Neural Information Processing,
D-89069 Ulm, Germany

Abstract. Robot recognition is a very important point for further improvements in game-play in ROBOCUP middle size league. In this paper we present a neural recognition method we developed to find robots using different visual information. Two algorithms are introduced to detect possible robot areas in an image and a subsequent recognition method with two combined multi-layer perceptrons is used to classify these areas regarding different features. The presented results indicate a very good overall performance of this approach.

1 Introduction

Due to the huge variety of robot shapes and designs in ROBOCUP middle size league vision based robot detection is a challenging task. A robot recognition method has to be very flexible to identify all the different robots but at the same time highly specific, in order not to misclassify similar objects outside of the playing field (e.g. black dressed children). Therefore many teams still recognize robots in their sensor data only as obstacles for collision avoidance.

At the same time, a good performance in ROBOCUP depends more and more on complex team behaviour. It is no longer sufficient, for a robot to localize itself on the playing field and behave autistically without taking care of other team robots. It is rather necessary that the robots act as and interact within a team. Furthermore, recognizing and tracking of opponent robots becomes desirable to improve the team strategy.

Robot interaction is definitively only possible if the relative position of the partner is known exactly. Otherwise e.g. a pass may fail and an opponent robot can take possession of the ball. Regarding the bad experiences of past ROBOCUP competitions, it is also risky, if the robots solely base their decision on the shared and communicated absolute position on the field, because the communication may fail or the robot doesn't know its own position exactly (or even may be totally wrong). With respect to this, it is quite clear, that it is necessary for a robot to detect and recognize other players by himself without sharing position information explicitly. Whereas this might be bypassed with better selflocalization and a fault tolerant communication equipment, there are other tasks like for example to dribble around opponent robots and to plan a path without colliding

with any obstacle along this path. There is no way to do so without any kind of detection of opponent robots.

A visual robot detection method for the ROBOCUP environment has to weight carefully two opposing goals. It has to be reliable and computationally inexpensive at the same time. This requires on the one hand the use of good, significant features, on the other hand the computational complexity that is needed to calculate these features needs to be low. To detect the robots during a game, the whole recognition task lasting from recording the image to the last decision step mustn't take any longer than a few milliseconds.

In contrast to the classical object recognition problems, there are only little restrictions about the robots shape and spatial dimensions. Every team have their own, sometimes very different robots, ranging from large and heavy almost quadratic cubes to highly dynamic, small and fragile ones. So there is need for a highly flexible, yet fast method to find in a first step the possible robot positions within the image, because it is only computational maintainable to process a subset of each image. Another important point is, that the method has to be fast and easily adaptable to new and unknown robot shapes.

In this paper we present a robot detection system using neural networks which extracts (possibly multiple) robots from a recorded image. To be able to handle the special requirements of the robot recognition in ROBOCUP the algorithm is split up into three independent subtasks. The first task is finding the possible robot positions in the original image (i.e. defining one or multiple region of interests (ROI)). The next step is to extract the features from these ROIs for further classification. The final classification decision is then performed on the basis of the extracted features by two neural networks. The first two steps are always the same (means they are not adapted to specific robots or environments). Only the neural network may be adapted on the current situation (e.g. a completely other robot shape) if required. It may even be possible to do this adaption within shortest time e.g. on-site before competitions.

The paper is organized as follows: Section 2 first explains the robot detection task as a whole. After a small introduction into the ROBOCUP scenario in 2.1 the following subsections 2.2–2.5 explain the individual steps in detail. Experiments and results are presented in section 3. In section 4 this paper is discussed in the context of related work. Finally section 5 draws conclusions.

2 Method

In this section the individual steps of the presented robot recognition method are explained in more detail. The itemization of the different steps is as follows:

1. Detect the region of interest,
2. extract the features from the RIOs,
3. classification by two neural networks,
4. arbitration of the classification results.

Figure 1 illustrates the data flow up to the end result. The first step (1) shows the detection of the regions of interest. We present two alternative methods

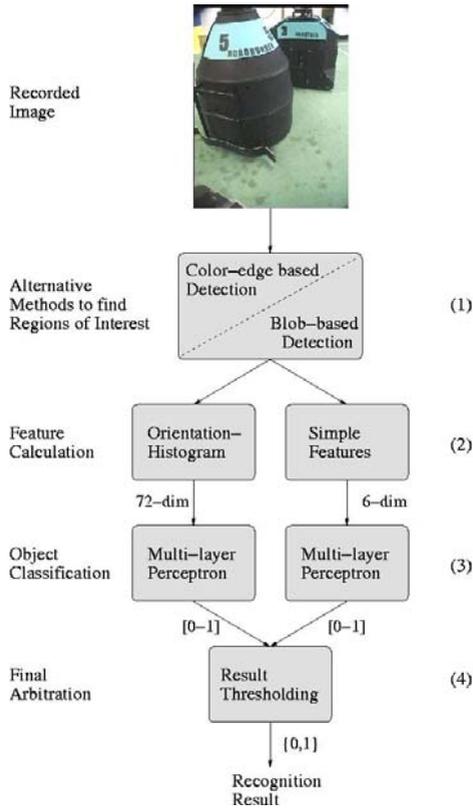


Fig. 1. Data-flow from image to recognition result

for finding them, one time a color-edge based one, the other time a color-blob based algorithm. Potential robot positions are searched here. For every possible area a data vector is calculated which includes orientation histograms and other, simpler features (2). These vectors are submitted to two artificial neural networks (3) and the results are then passed over to a final arbitration instance (4).

These are clearly separated tasks, so if there are robots that are not recognized well enough a customization of one of the networks (using the orientation histogram) can be applied. The second network only uses features described and predetermined by the ROBOCUP rules. As it is very important that the whole task is fast enough to process the images in "real time" (i.e. in an adequately short time) and is flexible enough to be adaptable to different opponents from game to game, in the following, each step is examined with respect to this.

2.1 The RoboCup Scenario

For those of the readers not familiar with the ROBOCUP scenario we first want to give you a short introduction. In the ROBOCUP middle size league four robots (three field player plus one goal keeper) playing soccer against four other ones.

All relevant objects on the field are strictly color coded: the ball is orange, the floor is green with white lines on it, the goals are yellow and blue, corner posts are blue and yellow colored and robots are mostly black with magenta or cyan color markers on it. During ROBOCUP championships there are also spectators around the field that can be seen from the robots.

There are also constraints about the robots size. Robots are allowed to have a maximal height of 80cm and a maximal width (resp. depth) of 50cm. Additional shape constraints and technical and practical limitations further restrict the possible robot appearances.

The game itself is highly dynamic. Some of the robots can drive up to 3 meters per second and accelerate the ball even higher. To play reasonably well within this environment at least 10–15 frames must be processed per second.

2.2 Region of Interest Detection

The first step is to direct the robots attention to possible regions within the recorded images. This is necessary because the feature calculation might be computational expensive and most time, large areas of the taken pictures are not of any interest. On the other hand, only a robot that stay within one of detected regions of interest can therefor be recognized later because all subsequent processing steps rely on this decision. So this attention control has to be as sound and complete as possible, to get all potential robot positions and not having to examine too much uninteresting areas. Two different algorithms are presented for this task each having its own quirks and peculiarities that score differentially in terms of speed and selectivity, so depending on the available computing power one of them may be used, or both may be mixed up in some way. Both methods rely on assertions made on the robots color and shape (as described in section 2.1) and therefore are rather specialized in the ROBOCUP scenario.

Histogram Method. The first approach examines all black areas within the picture. As it is currently save to assume, that each object on the playing field resides on the floor, we can calculate the size of the found regions easily. Regarding the already mention fact, that the robot size is restricted to a maximal size and that for now all robots in ROBOCUP middle size league are at least 30cm wide, all regions that do not achieve these restrictions are filtered out.

The blob detection is based on vertical and horizontal color histograms. To get the right position and size of an black blob we split the image in deliberate sub-images because one occurrence histogram of the black color for each direction may be not well-defined. By subdivision of the original picture to several areas it is easy to manage a detection of the interesting areas. Figure 2 shows the problem with histograms for several large areas.

The used image splitting is shown in Figure 3. It is assumed that all robots stand on the floor so the sub-images describe nearly the dimensions of the robots. Finally the histograms of the sub-images are subsequently searched for potential robots.

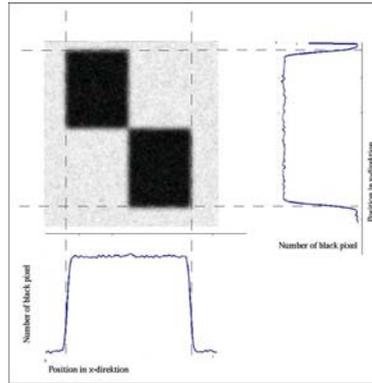


Fig. 2. Problems with the position-detection of color-blobs

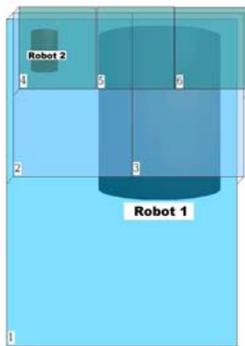


Fig. 3. Sub-images to check for black blobs

Color Edge Method. The second algorithm looks for black/green transitions within the picture starting at the bottom of the image, that indicate edges between (black) robots and the (green) floor. After calculating the real width of these lines and again filtering them with the minimal and maximal size constraints, the team color-makers above this lines are searched. This method already recognizes most robots reliably and selectively as long as they stand alone. Also, due to the high selectivity this method is much less robust against partially occlusions.

2.3 Feature Calculation

In the second step different features are calculated for the defined regions of interest. The different features describe different attributes of the robot. As the robot form cannot be predicted exactly, the features must be general enough to be applicable for different robot shapes, on the other hand specific enough to mask out uninteresting objects reliably. Beside that, the overall object detection

is done with a combination of all used features as explained in section 2.4. The features used are the following:

- Size of the black/green transition lines.
- Percentages of black (robot) and cyan/magenta (team-marker) color.
- Entropy.
- Orientation histograms.

The features are calculated from the original picture and from a segmented image. The segmented image is created by assigning each real color to a color class describing one of the above mentioned (section 2.1) object classes on the playing field or to a catch-all class that is not mentioned further on (described in detail in [1]). The first three feature types mostly check attributes asserted by the rules (e.g. size, color, color-marker). However the orientation histograms contain indirect details on the shape of the robot in a rather flexible way, but are strongly dependent on the right selected region. If the window is too large, not only the orientation of the robot but also of the background is calculated and examined. Vice versa, if the window is too small, we may overlook important parts of the robot.

The size of the black/green transition line shows the visible robot size near the floor or the width of the ROI depending on the used attention control method.

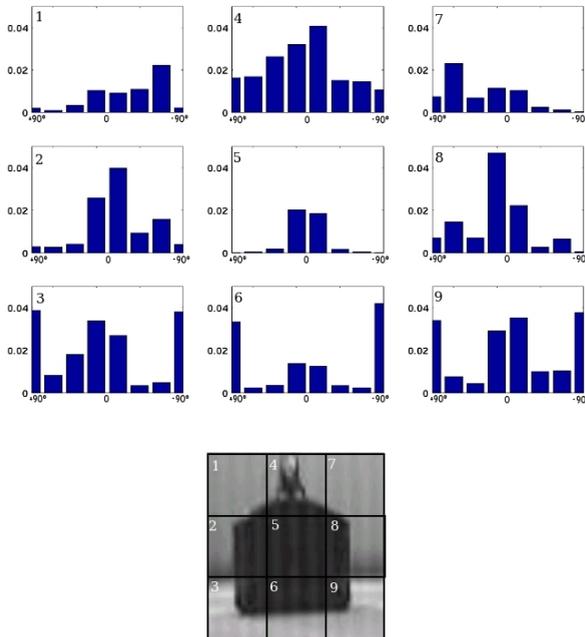


Fig. 4. Nine orientation histograms for one robot

The percentages of colors tells the team membership (team-maker) of a robot. A more general feature is the entropy. It is a indicator for the disorder within the area. A robot area is more disordered than a picture of the floor regarding the gray-scale values. Also a picture with the whole field is more disordered than a robot picture.

In Figure 4 the orientation histograms for one robot is shown. The histogram is made by accumulate the gradients in x and y direction detected by two Sobel filters on a grey image weighted by their quantity. The histogram is discretized into (in our case) eight chunks. Note that the histograms is calculated independently for nine sub-images, where the individual areas overlap of around 25%. This way, the orientation histograms are more specific for different prevailing edges within different parts of the image. In histogram number eight you can see e.g. the dominating vertical edge within the sub-image represented by the peak in orientation zero. In opposite, the horizontal bottom line of the robot is represented by another peak with orientation 90 degree in histogram number six. So the orientation histograms are a very flexible way to specify the robots shape within the region of interest.

2.4 Neuronal Networks Classification

Two neural networks do the actual classification in the whole robot recognition task. The networks are standard multi-layer perceptrons that are trained with a backpropagation algorithm as proposed e.g. in [2]. Both networks contain one hidden layer and a single output neuron. The first network gets only the data from the orientation histogram, the second one is fed with the other features. Both networks produce a probability value that describe its certainty of seeing a robot regarding the given input vector. To gain continuous output signals, sigmoidal functions are used in the output layer. The error function used for the backpropagation algorithm is the sum over the squared differences between the actual output value and the desired teaching pattern. Splitting the networks proved to be necessary, as otherwise the pure amount of orientation values suppress and outperform the other, simpler measurements.

The results in section 3 are generated with a training set of different pictures with and without robots. The resulting regions of interest are labeled manually to produce the best possible performance.

2.5 Arbitration

The final classification decision is made from a combination of the outputs of the two neural networks. Because every network only works on a subset of the features, it is important to get an assessment as high as possible from each individual network. Of course a positive feedback is easy, if both network deliver an assessment of nearly 100%. But in real life, this is only rarely the case. So the network outputs are rated that way, that only if both networks give a probability value bigger than 75%, it is assumed that a robot is found within the region of interest.

3 Experimental Results

In this section the results of the individual steps of the robot recognition task are described and discussed in detail. All tests are made using a training set of about 88 images. The size of the images is always PAL/4, i.e. 384×288 pixels. The images are taken with 3 different robots, containing 99 occurrences of them. The robots are recorded from different perspectives and from different distances. The teacher signal for training the neural network (i.e. the resulting bounding box around the found robot) is added by hand to assure the best possible performance. After the training phase the networks are able to calculate a classification from "robot" to "no robot" in the sense of a probability measure between zero and one.

The first two sections compare the different methods to calculate the regions of interest for the robots within the recorded images. After that, the overall performance is evaluated and finally the adaptation on new robots is discussed.

Because computational complexity and therefore the needed calculation time is very important in such a dynamic environment like the ROBOCUP, we measured the time needed for the individual processing steps on a 1.6 GHz Pentium4 mobile processor. These values are certainly highly implementation dependent, but may give an impression about how fast the whole object recognition task can be done.

3.1 Blob-Detection Method

The blob-detection method uses a more universal approach and detects all black areas within the picture. In our case, blob detection is simply performed by computing occurrence histograms in both vertical and horizontal direction for the color class of interest. Resulting blobs are then filtered with the size constraints found in the ROBOCUP rules. No further restrictions are taken into account apart from the size of the robots and the black color. As a result, this method detects more potential robot positions which implies more work in the subsequent two processing steps. On the other hand this method recognize all whole robots and more of the covered ones. The lower left image in Figure 5 shows, that overlapping objects have less influence on the detection performance than for the other described method. In the upper left image a false prediction can be seen (note the second white square in contrast to the upper right image).

This attention control method finds 93% (i.e. 92 of 99) of the robots within the images correctly. Additional 57 positions are found that do not contain a robot, but several other black objects. All robots which are not covered by other objects are detected. The accuracy of the robot detection is sometimes better compared to the other method as can be seen in the lower row of Figure 5.

On an average the method needs less then 1 ms to examine the whole picture and to detect all the possible robot positions. As a drawback of its flexibility, the following processing steps may take significantly more time than with the below mentioned attention control process due to the many false positives.

3.2 Black/Green Transition Detection Method

Using the attention control algorithm searching for black/green transitions within the image, the robots have to be "rule-conform" to be detected, otherwise they aren't selected or filtered out by the heuristics of this method. In the lower right image in Figure 5 you can see the consequence, if the robots size is determined by its bottom line only. If it is partially masked by another object, only parts of the robot may be used as region of interest or even filtered out because of the applied size assumptions.

This attention control method finds 92% of the robots within the images correctly. This means that 91 of the 99 robots are recognized that way, that a human expert marked them as sufficient. Additional 14 positions are found that do not contain a robot, but several other black objects. Again all robots which are not covered by other objects are detected. Missed robots are mostly far away and are hard to recognize in the image even for the human expert.

The advantage of this method is its speed and the low amount of wrong false classified areas, which again saves time in subsequent processing steps. On an average the method needs clearly less then 0.5 ms to examine the whole picture and to set the regions of interest.

3.3 Feature Calculation and Neural Processing

The time needed for the calculation of all features depends on the amount of found regions, as well as the implementation of the filters themselves. The used

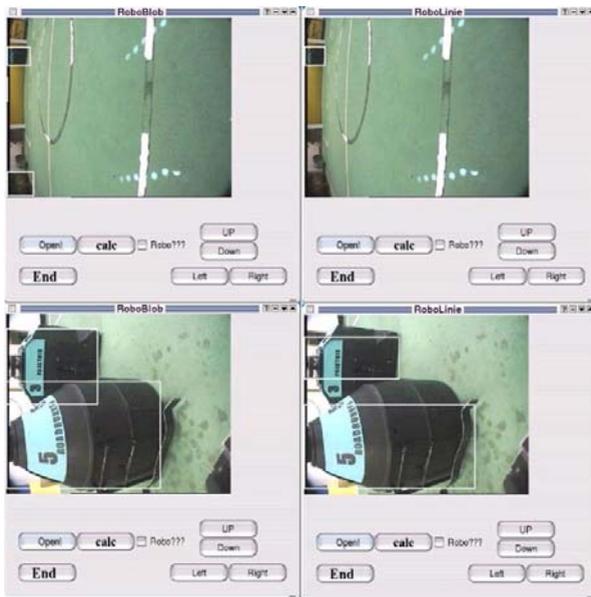


Fig. 5. Results using the different attention control algorithms

algorithms need around 174–223 ms depending on the size of the found region. Nevertheless the first attention control mechanism using the green/black transition needs significantly less overall processing time compared to the other, blob-based method because of the less false positives found. A preliminary, highly optimized version for the calculation of the orientation histogram (which consumes clearly the most time of the whole processing step) needs about 23 millisecond if applied to the whole (384×288 pixels) image.

The artificial neural networks are trained with around 200 feature vectors, about 40% of them contained data from real robots, the others are randomly chosen regions from the image. The final result of the neural networks again depend on the results delivered by the used attention control mechanism. Using the green/black transition method the overall correct robot recognition is about 95.3% regarding the delivered ROIs. When using the other (blob-based) algorithm, the result slightly decrease to 94.8%. This evaluation of the neural networks again is quite fast and uses clearly less than a millisecond.

3.4 Adaption and Retraining

If playing against robots with a totally different shape than that, used in the present training set, the network for the orientation histograms are likely to need adaption to the new situation. For this new training of the network, images of the new robots are needed. It is important to use images from different points of view and different distances. As the used learning algorithm is a supervised training method, the images have to be prepared so the precise robot positions are known. Now the network can be retrained for the orientation histogram. After a short time (around 1–2 minutes), the network is again ready to work.

Future work will focus on automating the training phase at the beginning of a game. Before a game-start only robots should be on the playing field, so every robot of the own team could take some pictures of the opponents which should fulfill the desired variety in orientation angle and distance. Herewith a reliable extraction should be possible and the learning of the new robot shape may be fully autonomous.

4 Related Work

Object detection is a well known problem in current literature. There are many approaches to find and classify objects within an image, e.g. from Kestler [3], Simon [4] or Fay [5] to name just a few that are developed and investigated within our department.

Within ROBOCUP the problems are rather less well defined then in their scenarios and real-time performance is not an absolute prerequisite for them, which may be the main reason that up to now there are only few workings are published about more complex object detection methods in ROBOCUP. Most of the participant within the ROBOCUP middle size league use a mostly color based approach, like e.g. in [6][7][8]. One interesting exception is presented from Zagal et. al. [9]. Although they still use color-blob information, they let the

robot learn different parameters for the blob evaluation, like e.g. the width or the height of the blob using genetic algorithms. Therewith they are able to even train the robot to recognize multi-colored objects as used for the beacons on both sides of the playing field (as used within the Sony legged league, which is rather comparable to the middle size league).

One attempt to overcome the limitations of pure color based algorithms is presented from Treptow et. al. [10] in which they trained an algorithm called adaboost using small wavelet like feature detectors. They also attach importance to let the method work reliably and with virtual real-time performance. Another approach, that even don't need a training phase at all is presented from Hanek et. al. [11]. They use deformable models (snakes), which are fitted to known objects within the images by an iterative refining process based on local image statistics to find the ball.

5 Conclusions and Future Work

Considering all the mentioned boundary conditions, robot recognition in ROBO-CUP middle size league is a difficult task. We showed, that splitting the problem into several subtasks can made the problem controllable. The combination of relatively simple pre-processing steps in combination with a learned neural decision entity results in a fast and high-quality robot recognition system.

We think, that the overall results can be even increased with a temporal integration of the robots position as we use it already for our self-localization [12] and described by other teams [10][13]. So partially occluded robots can be detected even if the robot is not detected in every single image. Future work is also planned on the selected features. With highly optimized algorithms there is again computing power left over, that can be used to increase the classification rate.

We collected a huge amount of real test images during a workshop with the robot soccer team from Munich. So we will focus on doing a very detailed investigation, how this method behave for all the extreme situations that can happen in ROBOCUP, like e.g. occlusion, or robots at image boundaries. It's also of interest, how the neural networks behave, if they are confronted with opponent robots not yet in the training images data base.

Acknowledgment

The work described in this paper was partially funded by the DFG SPP-1125 in the project *Adaptivity and Learning in Teams of Cooperating Mobile Robots* and by the MirrorBot project, EU FET-IST program grant IST-2001-35282.

References

1. Mayer, G., Utz, H., Kraetzschmar, G.: Playing robot soccer under natural light: A case study. In: RoboCup 2003 International Symposium Padua (to appear). (2004)

2. Russell, S.J., Norvig, P.: Artificial Intelligence: A Modern Approach. Prentice Hall, Upper Saddle River, NJ (1995)
3. Kestler, H.A., Simon, S., Baune, A., Schwenker, F., Palm, G.: Object Classification Using Simple, Colour Based Visual Attention and a Hierarchical Neural Network for Neuro-Symbolic Integration. In Burgard, W., Christaller, T., Cremers, A., eds.: Advances in Artificial Intelligence. Springer (1999) 267–279
4. Simon, S., Kestler, H., Baune, A., Schwenker, F., Palm, G.: Object Classification with Simple Visual Attention and a Hierarchical Neural Network for Subsymbolic-Symbolic Integration. In: Proceedings of IEEE International Symposium on Computational Intelligence in Robotics and Automation. (1999) 244–249
5. Fay, R.: Hierarchische neuronale Netze zur Klassifikation von 3D-Objekten (in german). Master's thesis, University of Ulm, Department of Neural Information Processing (2002)
6. Jamzad, M., Sadjad, B., Mirrokni, V., Kazemi, M., Chitsaz, H., Heydarnoori, A., Hajiaghahi, M., Chiniforooshan, E.: A fast vision system for middle size robots in robocup. In Birk, A., Coradeschi, S., Tadokoro, S., eds.: RoboCup 2001: Robot Soccer World Cup V. Volume 2377 / 2002 of Lecture Notes in Computer Science., Springer-Verlag Heidelberg (2003)
7. Simon, M., Behnke, S., Rojas, R.: Robust real time color tracking. In Stone, P., Balch, T., Kraetzschmar, G., eds.: RoboCup 2000: Robot Soccer. World Cup IV. Volume 2019 / 2001 of Lecture Notes in Computer Science., Springer-Verlag Heidelberg (2003)
8. Jonker, P., Caarls, J., Bokhove, W.: Fast and accurate robot vision for vision based motion. In Stone, P., Balch, T., Kraetzschmar, G., eds.: RoboCup 2000: Robot Soccer. World Cup IV. Volume 2019 / 2001 of Lecture Notes in Computer Science., Springer-Verlag Heidelberg (2003)
9. Zagal, J.C., del Solar, J.R., Guerrero, P., Palma, R.: Evolving visual object recognition for legged robots. In: RoboCup 2003 International Symposium Padua (to appear). (2004)
10. Treptow, A., Masselli, A., Zell, A.: Real-time object tracking for soccer-robots without color information. In: Proceedings of the European Conference on Mobile Robotics (ECMR 2003). (2003)
11. Hanek, R., Schmitt, T., Buck, S., Beetz, M.: Towards robocup without color labeling. In: RoboCup 2002: Robot Soccer World Cup VI. Volume 2752 / 2003 of Lecture Notes in Computer Science., Springer-Verlag Heidelberg (2003) 179–194
12. Utz, H., Neubeck, A., Mayer, G., Kraetzschmar, G.K.: Improving vision-based self-localization. In Kaminka, G.A., Lima, P.U., Rojas, R., eds.: RoboCup 2002: Robot Soccer World Cup VI. Volume 2752 / 2003 of Lecture Notes in Artificial Intelligence., Berlin, Heidelberg, Germany, Springer-Verlag (2003) 25–40
13. Schmitt, T., Hanek, R., Beetz, M., Buck, S.: Watch their moves: Applying probabilistic multiple object tracking to autonomous robot soccer. In: Eighteenth National Conference on Artificial Intelligence, Edmonton, Alberta, Canada (2002)