Neural Robot Detection in RoboCup

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

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

Abstract. Improving the game play in ROBOCUP middle size league requires a fast and robust visual robot detection system. The presented multilevel approach documents, that the combination of a simple color based attention control and a subsequent neural object classification can be applied successfully in real world scenarios. The presented results indicate a very good overall performance of this approach.

1 Introduction

In ROBOCUP middle size league, teams consisting of four robots (three field players plus one goal keeper) play soccer against each other. The game itself is highly dynamic. Some of the robots can drive up to 3 meters per second and accelerate the ball even more. Cameras are the main sensor used here (and often the only one). To play reasonably well within this environment, at least 10–15 frames must be processed per second.

Instead of grasping the ball and running towards the opponents' goal, team play and thus recognizing teammates and opponent robots is necessary to further improve robot soccer games. Because of noisy self localization and an unreliable communication between the robots, it is somewhat risky to rely only on the robots' positions communicated between the teammembers. In addition, opponent robots do not give away their positions voluntarily. Hence, there is a need for a visual robot detection system.

A method used for this task has to be very specific to avoid detecting too many uninteresting image parts, yet flexible enough to even detect robots that were never seen before. A robot recognition approach used in a highly dynamic environment like ROBOCUP also has to be very fast to be able to process all images on time.

In contrast to other object detection methods used on mobile robots, the scenario described here is more natural. Objects are in front of different, cluttered backgrounds, are often partially occluded, blurred by motion and are extremely variable in sizes between different images. Besides that, the shapes of robots from different teams vary highly, only limited by a couple of ROBOCUP regulations and some physical and practical constraints. Additional difficulties arise because the robots' vision system, which includes a very large aperture angle, causes extreme lens distortions.

The approach presented here is a multilevel architecture. This includes one level for the selection of interesting regions within the image that may contain a robot using simple and fast heuristics. The next level then classifies these regions with more complex and costly methods to decide if there is in fact a robot within the selected regions or not. This classification step is realized using artificial neural networks.

In this paper we focus more on the later steps: the choice of the used features, their parameterization and the subsequent neural decision making. We discuss in detail the performance of the neural networks on robots from the training set, on robot views previously unseen and even on totally unknown robot types. The test cases used for these results contain various kinds of perceptual difficulties as describe above. Considering this, the presented results indicate a good overall performance of the presented approach.

In the next Section the problem is explained. In Section 3 our method is described in detail, including all separate steps. Results are presented in Section 4. In Section 5 our results are discussed in the context of related work. Finally, Section 6 draws conclusions.

2 The Problem

As already mentioned, ROBOCUP is a quite realistic testbed. In contrast to other object recognition problems implemented on mobile robots, a lot more problems arise here that may also occur in real world scenarios: partial occlusions (between objects or at the image borders), vast size variations, huge variations in (robot) shapes, motion blur (through own or object motion), cluttered, unstructured and often unknown background and even unknown objects (e.g. robot types) that were never seen before. Figure 1 illustrates some of these problems, showing images recorded from one of our robots.

Fig. 1. Example images, taken from THE ULM SPARROWS, showing different types of perceptual difficulties. In the first two pictures (upper row) the variability in sizes of robots from the same type can be seen nicely. Recordings three and four showing occlusions, image five illustrates the lens distortion, which let the robot tilt to the left, and finally picture six is blurred by the robots own movement (the used pictures are admittedly not that badly blurred). Please note also, how different the individual robots look like.

3 Solution Approach

The robot recognition method we present in this paper can be roughly divided into the following individual steps:

1. Detect regions of interest.
2. Extract features from these regions.

3. Classify them by two neural networks.
4. Arbitrate the classification results.

The final result is a decision, if a robot is seen or not, and if it is part of the own or the opponent team. The whole process is illustrated in Figure 2 to further clarify the individual steps.

Fig. 2. Data flow from the source image, passing through the various processing stages, up to the recognition result.

In the first step, potential robot positions are searched within the recorded images to direct the robots attention to possibly interesting places (1). In the next step different features are calculated for each of the detected regions of interest (ROI – a rectangular bounding box). They describe different attributes of the robot as general as possible to be sensitive to the different robot shapes yet specific enough to avoid false positives (2). Used features are for example the width of the robot, a percentage value of the robots color and orientation histograms (and others described later in detail). All these features (i.e. the vector describing the features) are then passed over to two multilayer perceptron networks which vote for each feature vector if it belong to a robot or not (3). A final arbitration instance then decides wether we assume a teammate, an opponent robot or no robot at all (4).

3.1 Region of Interest Detection

In ROBOCUP, all relevant objects on the playing field are color coded. The floor is green, the ball is orange, the goals are blue and yellow and the robots are mainly black with a cyan or magenta color marker on top of them. So a simple attention control can be realized using a color based mechanism.

Consequently, region of interests are, in principle, calculated by a blob search within an segmented and color-indexed image (for a detailed explanation see e.g. [1] or [2]). Additional model based constraints reduce the number of found ROIs. Different possible methods are already explained in detail in [3], so we will not detail this here.

Of course, such a simple method cannot be perfect. There is always a tradeoff between how many robots are focused and how many regions not containing a robot are found. Because all subsequent processing steps only consider these found regions, it is important not to miss too many candidates here. On the other hand, too many non-sense regions increases the computational cost of the whole process.

3.2 Feature Calculation

The following features are finally used:

- width of the robot,
- percentage amount of black color within the whole ROI,
- percentage amount of (cyan or magenta) label color (separated into left, middle and right part of the ROI),
- an orientation histogram.

The first three items (respectively five values) are so called simple features (see Figure 2). The black and label color percentages are determined using the already mentioned color-indexed image.

Fig. 3. Orientation histogram using 3×3 subwindows and 8 discretisation bins. The individual histograms corresponds with the particular subwindows. For example a significant peak at direction zero degree, like e.g. in histogram eight, indicate a sole vertical line in the original image at the corresponding position.

The orientation histogram, the right process sequence in Figure 2, is a method to describe the overall shape of an object in a very general and flexible way. First of all, the found ROI is subdivided into several subwindows as illustrated in Figure 3. For each subwindow we then calculate independently the direction and the strength of the gradient in x- and y-direction (using a Sobel operator on a grayscale image). The directions of the gradients are discretized into a specific number of bins (e.g. eight in Figure 3). Finally we sum up all occurrences of the specific direction weighted by their strengths. To be even more flexible, the individual subwindows may overlap to a specified degree (see Section 4).

Because of extreme lens distortions, hollow robot bodies, or partial occlusions with other objects, there is a high chance that there is also a lot of noise inside the selected regions of interest. To minimize the influence of this, a binary mask is calculated from the robot and label colors with respect to the color-indexed image

(the mask is afterwards dilated several times to fill small holes). The orientation histogram is then only calculated, where the mask contains a positive value.

Fig. 4. Sketch of the various steps to calculate the histogram features including the robot mask and the maximal size reduction.

Another problem arises due to very different robot sizes. A close-up of a robot may contain a lot more details compared to a wide-distance image. So if a ROI is larger than a specific threshold, the ROI is scaled down to this upper size limit. For a more concise explanation of the various preprocessing steps, please have a look at Figure 4.

3.3 Neuronal Networks Classification

After calculating the different features, they are passed to two artificial neural networks as can be seen in Figure 2. One network only processes the simple features, the input for the second one are the orientation histogram values. The breakdown into two networks turned out to be necessary in order not to let the pure mass of data of the orientation histogram overvote the few simpler features.

Both neural networks are standard multilayer perceptron networks containing only one hidden layer and one single output neuron trained with a normal back-propagation algorithm. Both networks produce a probability value that describes the certainty of seeing a robot regarding the given input vector.

The input layer of the first network consists of 5 input neurons according to the five values described above. The number of neurons for the input layer of the second network depends on the parameterization of the orientation histogram. In Section 4 we present different parameterizations, but in general the size is the product of the number of subwindows in vertical and horizontal direction multiplied with the number of discretisation steps for the gradient direction. The number of neurons in the hidden layers is up to the network designer and is also evaluated in detail in Section 4.

3.4 Arbitration

The final classification decision is made from a combination of the outputs of the two neural networks. Because every network only uses 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 networks 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 above 75%, it is assumed that a robot is found within the region of interest. The membership of the robot to the own or the opponent team is determined using the robots color marker. If the ROI contains a lot more cyan pixels than magenta pixels or vice versa, the membership can be assumed for sure.

4 Experimental Results

As usual in a robot vision system, especially in combination with artificial neural networks, there are a lot of parameters that can and must be adjusted properly.

4.1 Parameterization

Having a closer look at the orientation histogram: The larger the overlap between the individual subwindows, the less problematic are ROIs that do not match a robot exactly (in size or in position), but on the other hand, the result is getting less specific. The number of subwindows in both directions controls how exact the shape representation will be. This is again a decision between flexibility and specificity. It can be wise to choose a larger number here if all robot types are known already, but disadvantageous, if more flexibility is needed. Exactly the same applies for the direction discretisation, i.e. the number of bins used in the histogram.

Another starting point for parametrization are the perceptron networks. While the number of input neurons is determined by the parameters of the orientation histogram (or the number of simple features) and the number of output neurons is fixed to only one (the probability for seeing a robot), there is still the hidden layer, which can be adjusted. The more neurons a network has in the hidden layer, the better it may learn the presented examples, but the higher is the risk to overfit the network and to lose plasticity and the ability to generalize the learned things. Another drawback of a large hidden layer is the increasing computational cost caused by the full connectivity between the neurons.

To be more concrete, the following values have been tested for the respective parameters:

- Subwindow overlap: 30%, 20%, 10% or 0% (no overlap).
- Number of subwindows: 2, 3 or 4 (in both directions).
- Number of histogram bins: 6, 8, 10, 12.
- Number of hidden neurons: identical to the number of input neurons or half, one third or one quarter of it (applies for the perceptron network which deals with the orientation histogram data, for the other network, see below).

4.2 Training

As this paper focuses on the evaluation of the object recognition performance and not the performance of the attention control or the overall process like in [3], we labeled all the images used in this section by hand. We used images from six different

types of robots with varying shapes collected during several tournaments (examples are shown in Figure 1). The collection contains robots from AIS/BIT-Robots (84 ROIs), Osaka University Trackies (138), Attempto Tübingen (116), Clockwork Orange Delft (148), Mostly Harmless Graz (102), Ulm Sparrows (94) and 656 ROIs as negative examples (164 carefully selected ones, like e.g. black trousers or boxes and 492 randomly chosen areas).

The learning phase is done by using five out of the six robot types, choosing 90% of the ROIs of each robot type, mix them with (again 90%) of the negative examples and train the artificial neural networks with them. The training phase is done three times for each combination of robot types and parameters to choose the network with the lowest mean square error regarding the training material to avoid unfortunate values of the (random) initialization. After that, we analyzed a complete confusion matrix on how the network recognizes the training data set itself or the remaining 10% of the robot (and negative) images which it has never seen before. Additionally, we calculated such a confusion matrix for a totally new (the sixth) robot type.

4.3 Classification Results

First results are provided for the neural network which is responsible for the orientation histogram values. In Table 1 the average results over all possible parameters for the different data sets (the training and the evaluation set) are listed. Because the neural network emits a probability value between zero and one, a positive answer is assumed, if the probability value is above 75%, i.e. ≥ 0.75 , as also used for the final arbitration step (see Section 3.4). The upper left cell displays the percentage of robots that are correctly identified as such, whereas the lower right cell represents the remaining percentage of regions that are assumed as not being a robot while in fact are a robot. The same applies for the other cells for regions not containing a robot.

First we test, how well the artificial networks perform if confronted again with the data set used during the training phase (the first part of Table 1). Note that the results are almost perfect, showing nearly 100% in the upper row. This means, that the networks are able to memorize the training set over a wide range of used parameters. The second part of the table is calculated using the evaluation data set. Remember that these values are made using images from the same robot types which are not presented to the network during the training phase. So even when averaging over all possible parameter values, the networks classified already over 90% correct.

Table 1. Resulting confusion matrices when presenting different data sets to the neural network responsible for the orientation histogram values. The results are averaged over all possible parameter permutations.

Training data	Positive	Negative
Correct decision	99.2%	99.8%
False decision	0.8%	0.2%
Evaluation data	Positive	Negative
Correct decision	91.9%	97.1%
False decision	8.1%	2.9%

Remember, that we trained the networks alternately only with five robot types. So now we can select the parameter combination, for which the networks perform best if confronted with the sixth type. Table 2 shows the performance for this settings one time for the evaluation data set and the other time for the sixth robot type (without negative examples respectively). The parameters proven to perform best are the following: Subwindow overlap 20%, number of subwindows 4, number of histogram bins 10, number of hidden neurons: 160 (identical to the number of input neurons). There are a couple of other parameter settings that perform only slightly worse.

Table 2. Resulting confusion matrices when using the best parameter settings for the orientation histograms. The results are averaged over all possible robot type permutations.

Evaluation data	Positive	Negative
Correct decision	100%	99.5%
False decision	0.0%	0.5%
Unknown robot type	Positive	Negative
Correct decision	96.4%	-
False decision	3.6%	-

Next results are shown for the artificial neural networks dealing with the simple features. The size of the hidden layer is the only variable parameter here. Testing all networks with a number of 3, 5, 10 and 15 hidden neurons, differences seems to be marginal, so we finally choose 3 neurons for this layer. Table 3 again presents the resulting value.

Table 3. Resulting confusion matrices for the network dealing with the simple features. The results are averaged over all possible robot type permutations.

Evaluation data	Positive	Negative
Correct decision	96.7%	100%
False decision	3.3%	0.0%
Unknown robot type	Positive	Negative
Correct decision	94.6%	-
False decision	5.4%	-

Finally we want to present one overall result. The parameters are chosen identically to the above ones. The network is trained with the following robot types: AIS/BIT-Robots, Osaka University Trackies, Attempto Tübingen, Clockwork Orange Delft and Mostly Harmless Graz (and the usual negative examples). As you can see in Table 4, the unknown robot type (THE ULM SPARROWS) is classified perfectly. The worst values are for the AIS/BIT-Robots where the network classifies 88% correctly and Tuebingen, where 9% of the robots are not recognized correctly.

4.4 Computation time

The vision system currently used in THE ULM SPARROWS provides 30 frames per second. Although it is not necessary to detect the robots in each single frame, a

Table 4. Resulting overall confusion matrices for one parameter set and one robot type combination.

Evaluation data	Positive	Negative
Correct decision	96.0%	100%
False decision	4.0%	0.0%
Unknown robot type	Positive	Negative
Correct decision	100%	-
False decision	0.0%	-

fast processing is an absolute prerequisite to be able to use the method in real tournaments. As absolute timing values should always been taken with a grain of salt, they may nevertheless give an impression on how fast the method can be. The values are measured on a 2.6GHz Pentium4 Processor, the image sizes are 640×480 Pixels. The various image preprocessing steps, like color indexing, mask building, observance of the upper size limit etc. need around 5ms, processing the orientation histogram averages 10ms and the final classification and arbitration step approximately 2ms. Note that some of the preprocessing steps like the color indexing need to be done anyway in the vision system for attention control of a lot of other objects.

5 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 [4], Simon [5] or Fay [6] 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 published about more complex object detection methods in ROBOCUP. Most of the participants in the ROBOCUP middle size league use color based approaches, like e.g. in [7][8][9]. One interesting exception is presented by Zagal et. al. [10]. 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. Thereby 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 in the Sony legged league, which is well comparable to the middle size league). Another method used in this league is introduced by Wilking et. al. [11]. They are using a decision tree learning algorithm to estimate the pose of opponent robots using color areas, its aspect ratio, angles between line segments and others.

One attempt to overcome the limitations of color based algorithms is presented by Treptow et. al. [12] where an algorithm called Adaboost uses small wavelet like feature detectors. Another approach, that does not even need a training phase at all, is presented by Hanek et. al. [13]. 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.

6 Conclusions and Future Work

The presented results indicate a good overall performance of our approach considering all the problematic circumstances mentioned above. We showed that splitting the problem into several subtasks (i.e. simple color based preprocessing in combination with neural network classification) made the problem manageable. The method appears to be a good basis for further improvements. Even though the images are taken during real tournaments, there are always surprises in real robotics. So additional, more specific features, or enhancements like temporal integration may further help to stabilize the overall detection rate (e.g. partially occluded robots can be tracked even if the robot is not observable or detected in every single image).

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.K.: Towards autonomous vision self-calibration for soccer robots. Proceeding of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS-2002) **1** (2002) 214–219
2. Mayer, G., Utz, H., Kraetzschmar, G.: Playing robot soccer under natural light: A case study. In: Proceedings of RoboCup-2003 Symposium. Lecture Notes in Artificial Intelligence, Berlin, Heidelberg, Germany, Springer-Verlag (2004)
3. Kaufmann, U., Mayer, G., Kraetzschmar, G.K., Palm, G.: Visual robot detection in robocup using neural networks. In: Proceedings of RoboCup-2004 Symposium (to appear). Lecture Notes in Artificial Intelligence, Berlin, Heidelberg, Germany, Springer-Verlag (2004)
4. 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
5. 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
6. Fay, R.: Hierarchische neuronale Netze zur Klassifikation von 3D-Objekten (in german). Master's thesis, University of Ulm, Department of Neural Information Processing (2002)
7. 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)
8. 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)
9. 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)
10. 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)
11. Wilking, D., Röfer, T.: Real-time object recognition using decision tree learning. In: Proceedings of RoboCup-2004 Symposium (to appear). Lecture Notes in Artificial Intelligence, Berlin, Heidelberg, Germany, Springer-Verlag (2004)
12. 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)
13. 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