

A Robust and Calibration-Free Vision System for Humanoid Soccer Robots

Ingmar Schwarz^(✉), Matthias Hofmann, Oliver Urbann, and Stefan Tasse

Robotics Research Institute, Section Information Technology,
TU Dortmund University, 44221 Dortmund, Germany
`ingmar.schwarz@tu-dortmund.de`

Abstract. This paper presents a vision system which is designed to be used by the research community in the Standard Platform League (<http://www.tzi.de/spl>) (SPL) and potentially in the Humanoid League (<http://www.robocuphumanoid.org>) (HL) of the RoboCup. It is real-time capable, robust towards lighting changes and designed to minimize calibration. We describe the structure of the processor along with major ideas behind object recognition. Moreover, we prove the benefit of the proposed system by assessing recorded image data on the robot hardware. The vision system has already been successfully employed with the NAO robot by Aldebaran Robotics (<http://www.aldebaran-robotics.com>) in prior RoboCup competitions as well as several minor events.

1 Introduction

This paper presents a vision system for humanoid robots in the RoboCup. Nowadays, it is common to ensure stable lighting conditions during competition games played at RoboCup. This allows for easy color separation and object recognition. Therefore, colortable-based image processors which impose either manual or semi-automatic calibration on users, are still widely utilized in the area. Assuming perfect manual color segmentation allows very simple recognition algorithms. These, however, become error-prone once the manual calibration is imperfect or sometimes even when it is done by a different person with a different bias towards labeling intermediate areas. As a result, such systems are limited in their adaptability and are considered inflexible.

The RoboCup organization forces the development in the area of autonomous robotics by applying rule changes to its various disciplines. This way, RoboCup is evolving towards a more realistic game play while improving individual skills of the robots. To this end, future games will likely be conducted outdoors in natural environments. This stresses the need for robust image processing systems capable of dealing with such challenging tasks.

The remainder of the paper is structured as follows: While Sect. 1.1 describes the objectives of the system, Sect. 1.2 presents related work on the topic. Section 2 outlines the means being used in the image processor, including field color and line detection as well as robot, ball, and goal post recognition. Section 3 proves the benefit of the proposed module in various experiments, i.e. real-game conditions.

We analyze the run time performance along with detection rates. We conclude our work in Sect. 4, and describe future work.

1.1 Objectives

In the following, we list our objectives for the proposed vision system. Our vision system is real time capable on the NAO robot, meaning it takes less than 33 ms to process both images, which is the time needed to capture the next images on the NAO robot. To save run time for the other tasks such as motion, localization and behavioral, we set our limit at 15 ms. This allows tracking moving objects such as the ball sufficiently fast for its application in RoboCup. We postulate the vision system to deal with *changing environmental conditions* such as lighting, indoor as well as outdoor. To ensure compatibility to other robots or hardware upgrades, the system should not be limited to run on a single hardware setup. The last main objective for our vision system is to reduce the needed calibration as much as possible.

1.2 Related Work

As we avoid the usage of calibration intensive color tables, our work regarding field color detection follows up on the work of Reinhardt [1] which we extended (see Sect. 2.1). Reinhardt introduced the first colortable-less vision system in the SPL of the RoboCup. For obstacle detection, there are different approaches in the SPL: Metzler et al. [2] proposed an algorithm for obstacle detection which recognize the feet of the robot. An alternative method from Fabisch et al. [3] relies very much on the jersey. To improve robot detection, we use the jersey and the body of the robot in addition to the feet. Our work suggests various heuristics to detect each of the relevant objects by providing specialized modules and verification steps. In contrast to Härtl et al. [4], we use dynamic thresholds for segmentation. Moreover, our approach is independent of colorimetric shift, and we use different algorithms for object detection. There are more approaches to deal with lighting differences and self-calibration, e.g. Hanek et al. [5], Bruce et al. [6], and Jüngel [7], but none of these approaches were tested to play outside in natural lighting conditions.

2 Vision System

The vision system is supplied by pictures from the cameras, and the current camera matrices. The term camera matrix refers to a transformation matrix between the projected middle point of the feet of the robot and the position of the respective lower or upper camera. This facilitates the utilization of 3-dimensional information, and is used to determine the position on the field of the detected objects relative to the robot. This section is structured as follows: The first step of the vision system detects the field color (see Sect. 2.1). After that,

the preprocessing step (see Sect. 2.2) extracts ball, line, and robot points, as well as goal segments. This information is required to compute and verify individual ball percepts (see Sect. 2.4), field lines with center circle (see Sect. 2.3), robots (see Sect. 2.6), and goal posts (see Sect. 2.5).

There is a trade-off between the robustness and reliability of the system, and the number of detected objects which might be false positives. The criteria when an object is deemed as detected have to be carefully determined. In general, we prefer to minimize the number of false positives. This mitigates the impairment of other vital functions of the system that depend on the results of the vision system, such as the localization and world modeling. Inaccuracies that influence the vision system have multiple causes: Due to the highly dynamic nature of a robot soccer game, robots are forced to look quickly at different points. This can result into motion blur, especially if a high exposure time is needed for low light environments. In addition, there are small differences in the cameras resulting in slightly different colors. Moreover, there is an interdependence between the vision system and the kinematics of the robot. Disturbances (e.g. caused by the walk of the robot) lead to inaccurate camera matrices. The vision system is therefore only executed when the camera matrix is plausible, i.e. the robot touches the ground with at least one sole.

2.1 Field Color Detection

Field color detection is the first vital step of the vision system as every object which has to be detected is located on the field, and thus the color is used to distinguish such objects from the field. Since light might change continuously, this step is repeated in each execution cycle. In many cases, the field color is not the most present color of the image as the robot might be located at the field border and looks outside the field. The SPL rules¹ define the field color as green-like. Our approach is similar to Reinhard [1]. We use a weighted color histogram on the YCbCr image from samples below the calculated horizon of the robot. Although the green part in the YCbCr colorspace is preferred by this weighting, other colors are accepted as well. The maxima of the weighted histograms of each channel are set as the current field color. We however do not use a fixed predefined color cube to cope with color changes induced by different lighting or cameras. The optimal color channel values from this step are used with a lighting- and color-dependent specific distance to classify a pixel as of field color throughout the vision system. These distances are taken from the histogram by using the width of the peak of the maximum. The field color is individually determined for each camera as those might differ significantly due to their different, automatically adjusting camera settings. To avoid field color changes when the robot looks out of the field, the field color is not allowed big changes once it is set.

¹ <http://www.tzi.de/spl/pub/Website/Downloads/Rules2014.pdf>.

2.2 Preprocessing

The main task of the preprocessor is to identify all possible object locations in the image by scanning the image along a fixed scan line grid. This allows the subsequent modules of the vision system to efficiently detect and verify the specific objects without scanning the image again. Inputs of the preprocessor are the previously computed field color (see Sect. 2.1), and the current camera matrix.

The process can be divided into three main steps: The first step uses *scan lines* which are then split into segments. To this end, only horizontal scan lines in the proximity of the horizon are used to find goal segments. The second steps *classifies* segments into field, line, ball, and unknown. The last step *filters and creates the data structures* needed by subsequent modules as the output.

The classification of field, ball and unknown segments is the core of the preprocessing and further described. The image is analyzed by means of horizontal and vertical scan lines that form a grid. The tightness of the grid can be configured by the user, significantly influencing the run time of the vision system (see Sect. 3.3). Moreover, the pixel tightness within the scan lines depends on the expected width of a line since a line is expected to be the smallest object to detect. Thus, the vision system is scalable and largely independent from the image resolution. The scan on each line stops when no field color is found and the segment exceeds the expected length of ball or line segments to save run time when multiple robots are in the image and to prevent detecting objects on these robots.

The scan process first splits the scan line at points of big color or brightness differences while counting pixels classified as of field color. The differences are measured at points where the direction of the color or brightness gradient changes to achieve robustness on blurred images. In the next step a segment is classified as of field color if at least 50 % is identified as field color. Afterwards, the segments are verified by their surrounding: Field segments are merged together and Line segments become verified line points when they are surrounded by field color, begin and end with leaps in brightness and if they do not exceed a specific width. Ball segments must begin and end with a color change using the cb- and cr-channel and are merged together after classification. Thus, it is noteworthy that no specific ball color is required.

All unidentified segments are then clustered to find the field border and points of interest for the robot detection. The upper image is only processed if the lower image does not detect a field border.

2.3 Line Detection

The detection of field lines plays a key role for the localization task as they are the most present feature on the field. The localization works best when the field lines and their widths are completely detected. The SPL rules state that field lines must be white and have a width of 5 cm. We distinguish between field lines and the center circle which has a radius of 75 cm. Hence, both types are processed differently in our vision system.

Edge detectors such as the Sobel and Roberts operators are commonly used for line recognition. The application of these algorithms is problematic in the SPL. Although Sobel is more robust, it is computationally expensive. Moreover, Sobel and Roberts are susceptible to motion blur and the quality of the detection decreases when lines are very thin. Hence, our approach connects the line points which have been computed in the preprocessing step instead scanning the image only in small parts to verify and extend field lines.

The line detection can be roughly divided into the following steps: First, we connect line points which are close to each other to form chains. Thus, we look for the point with the smallest Euclidean distance within the next two scan lines of the same type (horizontal or vertical). The line points from the preprocessing step are sorted by their scan line numbers for efficiency.

Second, *line segments* are built from such chains. Hence, we check the curvature of the chains in the image, and if it differs in one point too much from the other points, the segment is separated. The curvature as well as average and maximum errors (e_{avg} , and e_{max}) of a linear regression on the segment points are saved as meta information. We ensure that every point is assigned to exactly one line segment.

After the identification of line segments, we use their curvature to identify points on the center circle. All points on the center circle are projected on the field. We use the least squares fitting method [8] to find a circle by utilizing the set of the projected points (see Fig. 1). Moreover, we utilize an additional set of other line segments which are located in a suitable distance from the middle point to refine the circle.



Fig. 1. Center circle detection in a blurry image using line segments.

Field lines are formed from line segments which consist of a minimum number of points p_{min} , and are below a threshold regarding e_{avg} , and e_{max} . If segments have less than p_{min} points, we try to fuse the segment with an existing field line or other segment if it lies straight on and within a certain distance limit. Additionally, we check whether the field color is situated between the line segments. Lastly, we compute the width of a field line at its ends. The last step extends

the recognized field lines if possible. This is done by short scan lines which are looking for differences in brightness and for the field color at the end of the line.

2.4 Ball Detection

The ball is a unique feature on the field, and thus a relevant landmark for localization, i.e. for the resolving of the field symmetry [9]. The ball perception influences tactical decisions and accurate approaching. The vision system uses the shape, the color, and the size of the ball as relevant features. We further assume that the ball always touches the ground, i.e. a 2D ball model. Hence, the ball is at least partly surrounded by the field. Reflections on the ball surface, shadows, and partial occlusions make ball detection a challenging task during the game.

The official SPL ball has a diameter of 65 mm, is orange-like, and is originally used for Hockey. The preprocessing module provides the center of the ball segment that has been computed in the preprocessing step, and the average Y, Cb, and Cr values of the possible ball to ensure color-independence of this component. For ball detection, we utilize only the Cb, and Cr channels due to significant value deviations in the Y channel, e.g. due reflections of the spotlights on the ball.

Further, ball detection can be divided into *shape classification* and *validation*. In the shape classification step, we employ radiating scan lines beginning from the center of the ball segment to determine the outer points of the ball. The ball segment including position, length and average color, has been computed in the preprocessing step. The scan lines run until the border of the image is reached, a repeating deviation of the ball color is recognized, or the theoretical ball radius (computed with the camera matrix), is exceeded by factor 3. It is important that small deviations of the ball color are tolerated. The shape of the ball is calculated with the least squares fitting method [8] as in center circle detection. The segment is rejected if no circle can be built. The verification step deletes ball perceptions when size and shape of the ball is not plausible.

2.5 Goal Detection

We primarily use shape and color information for goal detection with the idea of color independence (no preconfigured colors) as in previous parts of the system. Output of the module is the position of each visible goal post (2 at most) along with a tag indicating the side (right or left). A goal post has a diameter of 10 cm, and is white as defined in the SPL rules². The base of the post is surrounded by field lines and the field color if not obstructed.

Input of the module is a set of extracted goal segments from the preprocessing. Goal segments consist of a start point and an end point as well as average values of the cb and cr channel of all pixels within the segment. Disturbances in

² However, the image processing system is evaluated on images from previous years which contain yellow-like goal posts.

the walk might cause distortions of the image's perspective. We therefore allow the goal post to be tilted. Consequently, as a first step, the angle of the goalpost in the image is determined. We connect the middle points of all goal segments to lines. All goal segments must have a consistent color and width, and we mainly use this kind of information to compute the upper and lower point of the posts. An additional set of horizontal scan lines is used to determine the width of the post which is vital for verification and is used to estimate the distance from the goal if the base of the goal post is not in the image. Moreover, if the upper point of the goal post is visible, the system extracts the crossbar to decide the side of the post.

2.6 Robot Detection

Reliable robot detection is very important in robot soccer as it allows for suitable path planning, and collision avoidance. Moreover, tactical decisions are often based on the positions of the robots on the field. Detected robots can be even used to support localization and world modeling [9].

In the SPL, all robots are of the same type. There are several possible approaches for robot detection like an exclusive search for team jerseys [3]. A major shortcoming of the approach is that it assumes the jersey or waistband always to be visible. In practice, jerseys might be occluded by robots, their body parts, and other objects.

Varying lighting conditions make it difficult to start with the jersey detection. Hence, we commence with feet detection, as the feet are usually surrounded by the field. Moreover, the feet of a visible robot should always be visible either in the upper or in the lower image. Input of the module is a set of obstacle points from the preprocessing step. Obstacle points are defined as points which could not have been assigned to any other object type.

We utilize scan lines to determine the width of the obstacle w_{scan} by looking for the field color in both horizontal directions for verification. The theoretical width of the object w_{opt} is computed with the camera matrix. We assume that the width of the obstacle at its feet is 21 cm which corresponds to the average width of the feet in any possible rotation. If the relation between w_{scan} and w_{opt} is too large, the obstacle is refused.

Lastly, color verification is performed. Since we assume the average walk height of the robot to be 53 cm, we look in suitable regions of the image for the jersey, and check for differences in the Cb and Cr channels to assess the team colors (which are red-like, and blue-like in the SPL). Moreover, we analyze whether the field color within the obstacle is more than 50% by applying diagonal scan lines. The last step checks for average deviations in the parts of the robots which are not covered by the jersey whether they differ too much from the color of the robot (which is white and gray in the SPL). The detection is easily configurable to be compatible with other colors. For instance, robots from the humanoid kid size are recognized when the main color of the obstacle is changed from white to black.

3 Evaluation

The examination of the proposed algorithm is twofold. On the one hand, we present and discuss its object recognition performance under various conditions (see Sect. 3.2). On the other hand, a run time analysis is conducted as real-time capability is of importance in highly dynamic environments (see Sect. 3.3).

3.1 Hardware

We use the NAO robot developed by Aldebaran Robotics in our experiments. The NAO is about 55 cm tall, and used in the SPL. It comes with an upper and lower camera with opening angles of 47.64° in vertical direction, and 60.97° in horizontal direction with no overlap. This comparatively small opening angles impose fast and excessive movement of the head which increases motion blur. We use the auto exposure, and white balance that is provided by the camera driver for evaluation. Aldebaran Robotics provides an API to configure the camera. A complete list and a more detailed description can be found on the supplier's web page³. This image processor has been evaluated with the NAO V4 version which offers limited computational power (Intel Atom 1.5 Ghz single core processor, and 1 GB RAM).

3.2 Object Recognition

In this section, we assess the performance of the system on in a quantitative and qualitative way. First, we examine the detection rates by using recorded image data from the robot hardware. The image log has been manually checked to get an idea whether objects in the image should have been detected by the system. This means that a set of detection criteria has to be defined for each type of object (see Table 1).

Table 1. Detection criteria of each object.

Object	Detection criteria
Field lines	- Line parts with a minimum length of 30 cm
Center circle	- At least a quarter of the center circle is observable
Ball	- At least a quarter circle of the ball is visible
	- The ball is located on the playground within the field borders
Goal posts	- The goal post is not cut by the left or right image border
	- The base of the goal post lies in the image
Robots	- The feet of the robot can be seen

³ <http://www.aldebaran-robotics.com/en/Discover-NAO/datasheet.html>.

The log file consists of 388 individual pictures that have been recorded at different places and conditions including the RoboCups and local Opens in the years 2008–2012 to check the performance of the algorithms in a realistic setting. This data is in the log file format of the B-Human framework [10], and available online.⁴ At least a third of the images are noisy and blurry, and many pictures contain persons and objects that do not belong to the ordinary field.

The log file consists of 388 individual pictures that have been recorded at different places and conditions including the RoboCups and local Opens in the years 2008–2012 to check the performance of the algorithms in a realistic setting. At least a third of the images are noisy and blurry, and many pictures contain persons and objects that do not belong to the ordinary field.

Table 2 shows the detection rates, false positives, and the number of entities that should have been recognized (N). It can be seen that the detection rates vary significantly depending on the type of object. We observe that goal post detection is most accurate with 86% of all objects that should have been detected. The low numbers of false positives are due to strict validation constraints in the heuristics of the algorithm. This leads to lower detection rates. As the vision system is operated at a maximum possible rate of 30 frames per second, we claim that the detection rates are sufficient for a successful application in a RoboCup game.

Table 2. Detection rates of the image processor and its components.

	Correct	False-positives	N
Field lines	61 %	8	601
Center circle	55 %	0	87
Ball	76 %	0	96
Goal posts	86 %	0	129
Robots	14 %	0	132

The system is able to detect the ball accurately even if less than half of the ball can be seen in the picture. Moreover, we detect balls that are approximately 4.5 m away from the robot (field size is 9×6 m) with the comparatively low resolution of 320×240 . If a higher resolution (e.g. 640×480 pixels) is used, the ball can be recognized on the complete field.

Figure 3 depicts a reliable goal detection with side assignments. If the robot stands, goals are detected even from a distance of more than 9 m. Due to the shape and color of the goal posts, and the correction mechanisms integrated in the vision system, our approach is able to compensate motion blur, and disturbances. The detection of the center circle is dependent on the distance to it and less reliable than the field line detection, since the curvature of the center circle is then less visible due to the low viewpoint of the NAO robot. Our robot

⁴ <http://www.nao-devils.de/downloads/IPLogs>.

detection does depend on the clustering of the preprocessor. If the other robot's distance is nearer, more scanlines intersect with that robot and consequently our detection rate increases. Robot's nearer than 2.5 m are detected with a success rate of more than 50 %.

The vision system has been successfully assessed in real-world conditions, e.g. the past three RoboCups as well as several events (see Fig. 2). To this end, it already worked in several outdoor environments, including for the shooting of the German crime series 'Tatort' without the need for calibration.

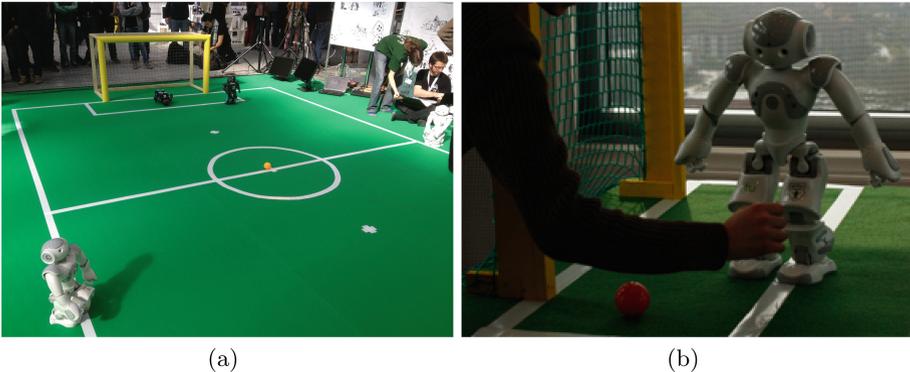


Fig. 2. Our robots playing at an exhibition in Zurich (a) and at the shooting for 'Tatort' (b).

3.3 Runtime Analysis

This experiment proves the real-time capability of the proposed system by recording the time needed to execute its various components. We reproduce situations that occur in regular RoboCup games to ensure the value of the assessment. To this end, a robot is placed on the field. Its task is to walk to the ball and kick it into the direction of the opponents' goal. Two robots are placed on the field, and the ball is rolled between the two robots. The experiment is conducted indoor in natural lighting conditions, i.e. no artificial light is turned on: This creates shadows and bright spots on the ground. The vision system is executed 1000 times in total which corresponds to 33 game seconds.

The run time is dictated by the preprocessing component which consumes more than 90 % of the total resources required as the majority of the scanning process takes place in this step. Thus, the number of used scan lines impose the overall run time. The algorithm can be easily scaled to fit users' requirements and platforms with different computational power. The current implementation allows the vision system to analyze both images in around 18 ms (17 ms for the pre-processing) on average using the highest camera resolution possible on the NAO robot, which is 1280*960 for the upper camera and 640*480 for the lower camera. The number of scan lines used is 92 for each image. If the resolution on both



Fig. 3. Goal posts with side assignments (a) and inaccurate detection of the left goal post due to motion blur (b).

cameras is halved, the run time decreases to around 14 ms total on average. The maximum run time in each case is only 2 ms higher while the minimum run time is at 5 ms if only the lower image is processed (i.e. the field border is detected in the lower image).

4 Conclusion and Future Work

This paper presents a real-time capable, robust, calibration-free, and easy configurable vision system that has been explicitly developed to meet the needs of the RoboCup Standard Platform League. To meet the run time requirements for the rest of the code, the number of scan lines is configurable which has been done throughout the past years. The concepts behind object recognition and the usage of scan lines can be slightly modified to allow their application in other domains. We have shown the benefits of the system in various experiments with the NAO robot, and stress that detection rates are sufficient for its application in RoboCup as the system is executed 30 times per second. At RoboCup 2014, we successfully performed a live game outside of the main hall with no calibration or adjustments to show the capabilities of our vision system. We further captured a video with experiments outside of our lab⁵. Throughout the last three years we adjusted the resolution for our cameras and the number of scan lines used and performed at several events (see Fig. 2) without the need to calibrate any further part of our vision system.

We deliberately avoid the employment of color tables due to their poor adaptability and robustness towards lighting changes. Instead, we use a variance of the field color detector proposed by Reinhardt [1]. The implementation is mostly independent from camera properties such as its resolution. This alleviates the

⁵ <https://www.youtube.com/watch?v=WXBEQF8k3b0>.

migration to other hardware platforms than the NAO. We exploit domain knowledge to achieve lighting-independence, and ensure robustness against motion blur and colorimetric shift. Since our vision system partly relies on the camera matrix, calibration is required for the best performance. For next year, we plan to calibrate this part in our vision chain while playing to completely remove the need for calibration. Our robots use auto settings provided by the cameras and no calibration of those settings was needed in the past years.

For the future, we suspect rule changes towards a more realistic color scheme of the goals and the ball in the near future, i.e. the ball used in the SPL might become a black and white pattern. Our goal is to extend the vision system to work with such amendments. Additionally, we aim at improving the performance of the system by refining the robot detection component in future work.

References

1. Reinhardt, T.: Kalibrierungsfreie bildverarbeitungsalgorithmen zur echtzeitfähigen objekterkennung im roboterfußball. Master's thesis, Hochschule für Technik, Wirtschaft und Kultur Leipzig (2011)
2. Nieuwenhuisen, M., Behnke, S., Metzler, S.: Learning visual obstacle detection using color histogram features. In: Röfer, T., Mayer, N.M., Savage, J., Saranlı, U. (eds.) RoboCup 2011. LNCS, vol. 7416, pp. 149–161. Springer, Heidelberg (2012)
3. Fabisch, A., Laue, T., Röfer, T.: Robot recognition and modeling in the robocup standard platform league. In: Pagello, E., Zhou, C., Behnke, S., Menegatti, E., Röfer, T., Stone, P. (eds.) Proceedings of the Fifth Workshop on Humanoid Soccer Robots in Conjunction with the 2010 IEEE-RAS International Conference on Humanoid Robots, Nashville, TN, USA (2010)
4. Visser, U., Röfer, T., Härtl, A.: Robust and efficient object recognition for a humanoid soccer robot. In: Behnke, S., Veloso, M., Visser, A., Xiong, R. (eds.) RoboCup 2013. LNCS, vol. 8371, pp. 396–407. Springer, Heidelberg (2014)
5. Hanek, R., Schmitt, T., Buck, S., Beetz, M.: Towards RoboCup without color labeling. In: Kaminka, G.A., Lima, P.U., Rojas, R. (eds.) RoboCup 2002. LNCS (LNAI), vol. 2752, pp. 179–194. Springer, Heidelberg (2003)
6. Bruce, J., Balch, T., Veloso, M.: Fast and inexpensive color image segmentation for interactive robots. In: Proceedings. 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2000), vol. 3, pp. 2061–2066. IEEE (2000)
7. Jüngel, M.: Using layered color precision for a self-calibrating vision system. In: Nardi, D., Riedmiller, M., Sammut, C., Santos-Victor, J. (eds.) RoboCup 2004. LNCS (LNAI), vol. 3276, pp. 209–220. Springer, Heidelberg (2005)
8. Chernov, N., Lesort, C.: Least squares fitting of circles. *J. Math. Imaging Vis.* **23**(3), 239–252 (2005)
9. Tasse, S., Urbann, O., Hofmann, M.: SLAM in the dynamic context of robot soccer games. In: Chen, X., Stone, P., Sucar, L.E., van der Zant, T. (eds.) RoboCup 2012. LNCS, vol. 7500, pp. 368–379. Springer, Heidelberg (2013)
10. Röfer, T., Laue, T., Müller, J., Bartsch, M., Batram, M.J., Böckmann, A., Bösch, M., Kroker, M., Maaß, F., Münder, T., Steinbeck, M., Stolpmann, A., Taddiken, S., Tsogias, A., Wenk, F.: B-human team report and code release 2013 (2013). <http://www.b-human.de/downloads/publications/2013/CodeRelease2013.pdf>